ICES REPORT 10-50

December 2010

Converting an Unstructured Quadrilateral Mesh to a Standard T-spline Surface

by

W. Wang, Y. Zhang, M.A. Scott, T.J.R. Hughes

The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: W. Wang, Y. Zhang, M.A. Scott, T.J.R. Hughes, "Converting an Unstructured Quadrilateral Mesh to a Standard T-spline Surface", ICES REPORT 10-50, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, December 2010.

Noname manuscript No.
(will be inserted by the editor)

Converting an Unstructured Quadrilateral Mesh to a Standard T-spline Surface

Wenyan Wang · Yongjie Zhang · Michael A. Scott · Thomas J.R. Hughes

Received: date / Accepted: date

Abstract This paper presents a novel method for converting any unstructured quadrilateral mesh to a standard T-spline surface, which is C^2-continuous except for the local region around each extraordinary node. There are two stages in the algorithm: the topology stage and the geometry stage. In the topology stage, we take the input quadrilateral mesh as the initial T-mesh, design templates for each quadrilateral element type, and then standardize the T-mesh by inserting nodes. One of two sufficient conditions is derived to guarantee the generated T-mesh is gap-free around extraordinary nodes. To obtain a standard T-mesh, a second sufficient condition is provided to decide what T-mesh configuration yields a standard T-spline. These two sufficient conditions serve as a theoretical basis for our template development and T-mesh standardization. In the geometry stage, an efficient surface fitting technique is developed to improve the geometric accuracy. In addition, the surface continuity around extraordinary nodes can be improved by adjusting surrounding control nodes. The algorithm can also preserve sharp features in the input mesh, which are common in CAD (Computer Aided Design) models. Finally, a Bézier extraction technique is used to facilitate T-spline based isogeometric analysis. Several examples are tested to show the robustness of the algorithm.

Keywords Unstructured Quadrilateral Mesh · Standard T-spline · Gap-free T-mesh · Extraordinary Node · Sharp Feature

Y. Zhang · W. Wang
Department of Mechanical Engineering, Carnegie Mellon University
Pittsburgh, PA 15213, USA
Tel.: (412) 268-5332
Fax: (412) 268-3348
E-mail: jessicaz@andrew.cmu.edu

T.J.R. Hughes · M.A. Scott
Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712, USA

1 Introduction

An important problem in Reverse Engineering is to construct 3D geometry from scanned imaging data. With the rapid development of 3D scanning data acquisition and automatic mesh generation techniques, the geometries obtained are always in the form of polygonal meshes. For example, Figure 1(a) shows an unstructured quadrilateral mesh of a human head. Besides polygonal meshes, splines are another widely used representation of freeform geometry, especially in Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), and Computer Aided Engineering (CAE). Splines can represent a wide range of geometric shapes accurately with high order continuity. Recently, a spline-based analysis method named "isogeometric analysis" was developed [7, 2], which has advantages over traditional finite element analysis. To improve geometric accuracy and continuity, and to achieve compatibility with CAD systems and isogeometric analysis, it is always desirable to convert polygonal meshes into continuous, high-order spline surfaces. In this paper, we focus on converting an arbitrary unstructured quadrilateral mesh into a standard T-spline surface with C^2-continuity everywhere except at local regions surrounding extraordinary nodes. The solution to this problem provides engineers with the opportunity to transition legacy bilinear quadrilateral surface meshes of engineering structures, which have been accumulated over the forty-year history of commercial finite element structural analysis programs, to T-splines, and to analyze them as such and compare results with traditional finite element technology. (We note that the LS DYNA program developed and marketed by Livermore Software Technology, Inc., already has the capability to compute with T-splines; see Benson *et al*. [3, 4].)

Considerable work has been devoted to converting triangular meshes to spline representations. In [1], an algorithm was developed to construct NURBS surfaces from triangu-

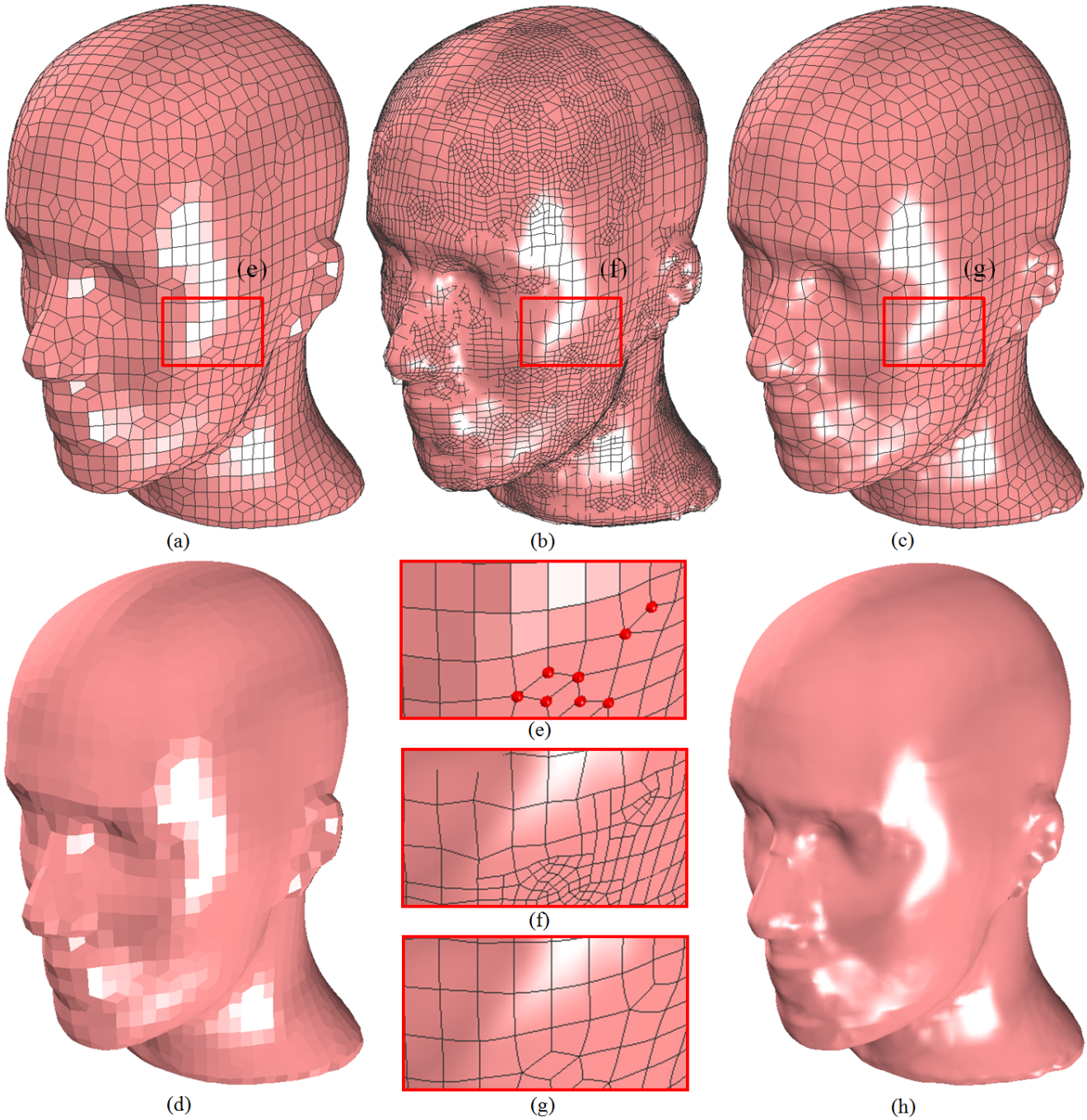


Fig. 1 A human head model. (a) The input unstructured quadrilateral mesh; (b) the constructed T-spline surface and T-mesh; and (c) the extracted Bézier elements. (d) and (h) are the shading results of the input mesh and the T-spline surface, respectively. (e) to (g) show details, and red points in (e) denote extraordinary points. There is a subtlety here that is difficult to see. The original mesh in (a) consists of bilinear quadrilateral elements. The T-spline is bicubic and C^2-continuous except in the vicinity of extraordinary points. The Bézier elements are embedded in the T-spline. Consequently, the Bézier elements and the original bilinear quadrilateral elements are different, despite appearances to the contrary.

lar meshes for organic structures. Based on Periodic Global Parameterization (PGP), another method was proposed for converting triangular meshes into T-splines [9, 8]. Starting from a polycube with the same topology as the input triangular mesh, Wang *et al*. [16] created a parametric map between the polycube and the geometry, and utilized this map to construct T-splines. In [20], a skeleton-based method was developed to construct solid NURBS for isogeometric analysis of arterial blood flow.

In this paper we develop a novel method to convert an arbitrary unstructured quadrilateral mesh to a standard bicubic T-spline surface, without consideration of input topology

or parameterization. For a standard T-spline, the basis functions provide a partition of unity [14]. There are two main stages in the conversion algorithm: the topology stage and the geometry stage. We take the input mesh directly as the initial T-mesh, and the topology stage aims to make the initial T-mesh gap-free and standard. Templates are designed for each type of quadrilateral element configuration to ensure it is gap-free and then additional nodes are inserted to standardize the T-mesh. Sufficient conditions are derived to guarantee the generated T-spline is gap-free around extraordinary nodes, and to decide what T-mesh configurations yield a standard T-spline. These two sufficient conditions provide a theoretical basis for the template development and T-mesh standardization. In the geometry stage, we improve accuracy and continuity by adjusting control nodes in the T-mesh. An efficient surface fitting technique is developed to make the output T-spline interpolate all the nodes in the input mesh. In addition, the conversion algorithm preserves sharp features present in the input model. Finally, Bézier elements are extracted from the constructed T-spline to facilitate isogeometric analysis [5, 12]. We applied the algorithm to several human anatomic models and CAD models with sharp features.

The remainder of this paper is organized as follows. Section 2 reviews T-splines. Section 3 presents an overview of the new algorithm and then the following sections explain the algorithm in detail. Section 4 discusses how to deal with extraordinary nodes in order to produce a gap-free and standard T-mesh. Section 5 describes sharp feature preservation. Section 6 explains surface fitting and continuity improvement. To facilitate isogeometric analysis, Bézier elements are extracted in Section 7. Section 8 presents some T-spline results, and Section 9 draws conclusions.

2 A Review of T-splines

T-splines [13] are a generalization of NURBS (Non-Uniform Rational B-Spline) [10]. A NURBS surface is defined by its degree, two global knot vectors and control points. The formula for a NURBS surface is

$$S_N(u,v) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} C_{ij} w_{ij} N_{i,d}(u) N_{j,d}(v)}{\sum_{i=0}^{m}\sum_{j=0}^{n} w_{ij} N_{i,d}(u) N_{j,d}(v)}, \tag{1}$$

where C_{ij} represents one control point, w_{ij} is the weight corresponding to C_{ij}, $N_{i,d}(u)$ and $N_{j,d}(v)$ are B-spline basis functions defined by two *global* knot vectors, $\mathbf{u} = [u_0, u_1, \cdots, u_{m+d}, u_{m+d+1}]$ and $\mathbf{v} = [v_0, v_1, \cdots, v_{n+d}, v_{n+d+1}]$, and d is the degree. For NURBS, the control points must lie topologically in a rectangular grid or mesh ($m \times n$ in this case). NURBS provide a unified geometry representation of conic sections and free-form shapes. They have been widely used in engineering design and manufacturing.

Unlike NURBS, T-splines allow T-junctions in their control grid. A T-junction terminates a row or column of control points in the control grid. T-splines have been applied to important problems in Computer Aided Geometric Design (CAGD) such as local refinement [14], NURBS merging [13], control mesh simplification [17], and trimmed NURBS conversion [15]. Unlike NURBS, they can also model geometry of arbitrary topological genus [13]. The formula for a T-spline surface is

$$S(s,t) = \frac{\sum_{i=0}^{n} w_i C_i B_i(s,t)}{\sum_{i=0}^{n} w_i B_i(s,t)}, \quad (s,t) \in \Omega, \tag{2}$$

where w_i is the weight for the control point C_i, $B_i(s,t) = N_i^s(s) N_i^t(t)$, N_i^s and N_i^t are B-Spline basis functions defined by two *local* knot vectors, $\mathbf{s}_i = [s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}]$ and $\mathbf{t}_i = [t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}]$ when degree $d = 3$, and Ω is the domain of the T-spline surface in parameter space. T-spline control points may not be connected in the form of a rectangular grid. A T-mesh is introduced to provide the connectivity of the control points. Each edge connecting two control points in the T-mesh is labeled with a ***knot interval***, which represents the parametric length of that edge. Two nodes[1] connected by one edge in the T-mesh have either the same s or t value. An edge whose two end nodes share the same t (or s) value is called an *s-edge* (or *t-edge*), respectively. The local knot vectors for each node can be inferred from the T-mesh. Two constituents are required to define one T-spline surface: the degree and the T-mesh. For NURBS, there are two *global knot vectors*, $\mathbf{u}$ and $\mathbf{v}$, but for a T-spline, each node C_i has its own *local knot vectors*, $\mathbf{s}_i$ and $\mathbf{t}_i$, which are inferred from the T-mesh. A T-mesh contains regular nodes, extraordinary nodes and T-junctions. A ***regular node*** has valence four, and an ***extraordinary node*** has valence other than four and is not a T-junction. NURBS can be treated as a special case of T-splines in which the T-mesh is simply a rectangular grid without T-junctions. T-splines generally use local parametric coordinate systems, especially when extraordinary nodes are involved. For a regular node or T-junction C_i, the local knot vectors are inferred from its T-mesh neighborhood by the following rule [14].

Knot Vector Inference Rule (for regular nodes and T-junctions): Suppose the knot coordinates of a regular node or a T-junction C_i are (s_{i2}, t_{i2}). Its two local knot vectors, $\mathbf{s}_i = [s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}]$ and $\mathbf{t}_i = [t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}]$, are determined as follows. Consider a ray in the parameter space,

[1] We use the terminologies "node" and "control point" synonymously.

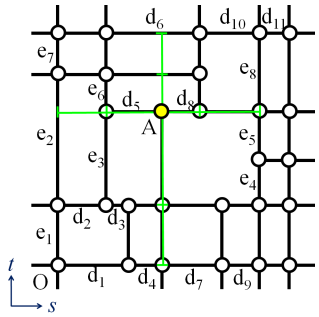


Fig. 2 A local region of a T-mesh. O corresponds to the parametric origin, d_i label the s-edge knot intervals, and e_i are the t-edge knot intervals.

$R(\alpha) = (s_{i2} + \alpha, t_{i2})$ where $\alpha \geq 0$, s_{i3} and s_{i4} are the s coordinates of the first two t-edges intersected by the ray (not including s_{i2}). Similarly, s_{i0} and s_{i1} are found when $\alpha \leq 0$. The other knot vector $\mathbf{t}_i$ can be inferred in a similar manner.

Taking node A in Figure 2 as an example, we suppose O corresponds to the parametric origin, then $(d_1 + d_4, e_1 + e_3)$ are the knot coordinates of A. The two local knot vectors of A are $\mathbf{s}_A$ and $\mathbf{t}_A$. For $\mathbf{s}_A$, $d_1 + d_4$ is the center knot. Considering a ray in the parameter space, $R(\alpha) = (d_1 + d_4 + \alpha, e_1 + e_3)$, where $\alpha \geq 0$, the first two t-edges intersected with the ray are the two right knots in the local knot vector $\mathbf{s}_A$. The left two knots can be obtained when $\alpha \leq 0$. Then we have $\mathbf{s}_A = [0, d_2, d_1 + d_4, d_1 + d_4 + d_8, d_1 + d_4 + d_8 + d_{10}]$. Similarly, we can obtain $\mathbf{t}_A = [0, e_1, e_1 + e_3, e_1 + e_2, e_1 + e_2 + e_7]$.

Sederberg *et al.* [14] introduced three types of T-splines: standard, semi-standard and non-standard. A ***standard T-spline*** is one for which

$$\sum_i B_i(s,t) = 1, \quad \forall (s,t) \in \Omega. \tag{3}$$

A ***semi-standard T-spline*** is one for which

$$\sum_i w_i B_i(s,t) = 1, \quad \forall (s,t) \in \Omega, \tag{4}$$

where not all $w_i = 1$. A ***non-standard T-spline*** is one for which no set of weights can be found for the blending functions to satisfy Equation (4).

In this paper we focus on converting an arbitrary unstructured quadrilateral mesh to a member of a restricted class of standard T-spline surfaces. In general, characterizing T-mesh configurations which yield a standard, semi-standard, or non-standard T-spline is an open problem.

3 Algorithm Overview

As shown in Figure 3, there are two stages in the conversion algorithm: the topology stage and the geometry stage. Input unstructured quadrilateral meshes may be generated using an octree-based iso-contouring method [18, 19, 11]

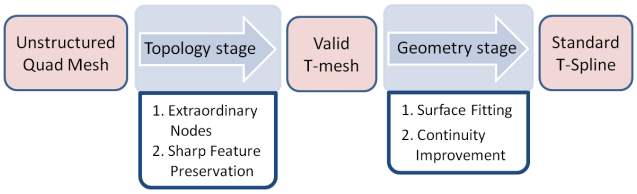


Fig. 3 An overview of the algorithm to convert an unstructured quadrilateral mesh to a standard T-spline surface.

and taken as initial T-meshes. The main goal of the topology stage is to construct gap-free and standard T-meshes, preserving sharp features present in the input mesh. In the geometry stage, we minimize the error between the input meshes and the output T-spline surfaces, and relocate the control points surrounding extraordinary nodes to improve surface continuity.

Topology Stage: The initial T-mesh has regular and extraordinary nodes. We can omit the topology stage if the input is a structured mesh and there are no sharp features. A structured mesh only contains regular nodes. For an unstructured mesh, the main work of this stage is to deal with extraordinary nodes and make the initial T-mesh valid. A ***valid T-mesh*** is one that yields a gap-free and standard T-spline surface. In the topology stage, we first classify the input quadrilateral elements into six categories, and for each category we design templates and insert additional nodes to guarantee the constructed T-mesh is gap-free and standard. To preserve sharp features, zero-length edges are inserted at the sharp edges and corners in the T-mesh. A ***zero-length edge*** is one edge with zero knot interval.

Geometry Stage: This stage aims to improve geometric accuracy and continuity. After we obtain a valid T-mesh, an efficient surface fitting technique is developed to improve the geometric accuracy by relocating control points. We can guarantee that the output T-spline surface interpolates all the nodes in the input mesh. Then the control nodes around each extraordinary node are adjusted to improve the local continuity. A standard T-spline surface is then constructed from the T-mesh, and Bézier elements are extracted for isogeometric analysis.

4 Extraordinary Nodes

If all the nodes in the input mesh are regular, the initial T-mesh is topologically correct. However, extraordinary nodes are unavoidable for complex geometry. Unlike regular nodes, the extraordinary nodes in the initial T-mesh need to be handled properly, otherwise they may introduce gaps in the T-spline surface. Moreover, the T-spline surface is no longer C^2-continuous around extraordinary nodes [13]. To generate a gap-free T-mesh, we first design templates for each type of quadrilateral element configurations and then derive

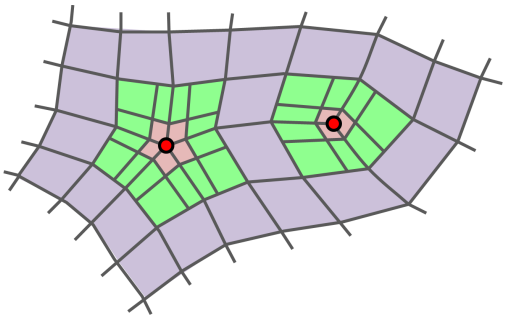


Fig. 4 Two conforming 2-ring neighborhoods around extraordinary nodes (red points) in a cubic T-mesh. Pink elements form the 1-ring neighborhoods and these elements together with the green elements form the 2-ring neighborhoods.

a sufficient condition to render the T-mesh gap-free. Then a sufficient condition is derived to determine when a T-mesh configuration is standard. These sufficient conditions provide a theoretical basis for the template development and T-mesh standardization. Finally, a knot vector inference rule is given for extraordinary nodes and their adjacent nodes, and a standard T-spline surface is constructed from the generated T-mesh. Here are some definitions we will need.

Definition 4.1. A ***quasi-uniform T-mesh*** is a T-mesh which contains edges with only zero or unit knot interval length.

Definition 4.2. A ***p-ring neighborhood*** around an extraordinary node is a set of T-mesh faces formed by first adding all the T-mesh faces which are coincident with the extraordinary node to the set. Then this set is expanded by adding all the T-mesh faces which are coincident with any T-mesh face already in the set. This process is performed p times.

Definition 4.3. A ***conforming p-ring neighborhood*** around an extraordinary node is a p-ring neighborhood where the set of T-mesh faces form a conforming quadrilateral mesh, that is, one in which there are no T-junctions.

By definition, a conforming p-ring neighborhood does not contain any T-junctions. Figure 4 shows two conforming 2-ring neighborhoods for one valence-3 node and one valence-5 node in the T-mesh. The pink faces are the 1-ring neighborhood and these faces together with the green faces form the 2-ring neighborhood. Note that each 1-ring or 2-ring neighborhood is a conforming quadrilateral mesh around the extraordinary node.

4.1 Template Development

There are six types of quadrilateral elements in the initial T-mesh: elements with none, one, two (neighboring or diagonal), three and four extraordinary nodes. We design two sets of templates for these six types of elements, which are listed in Table 1. Note in Table 1, all the red edges have zero knot interval and the grey domains have zero parametric area. The rule for the template development is that, for each extraordinary node, we make the T-mesh gap-free and standard.

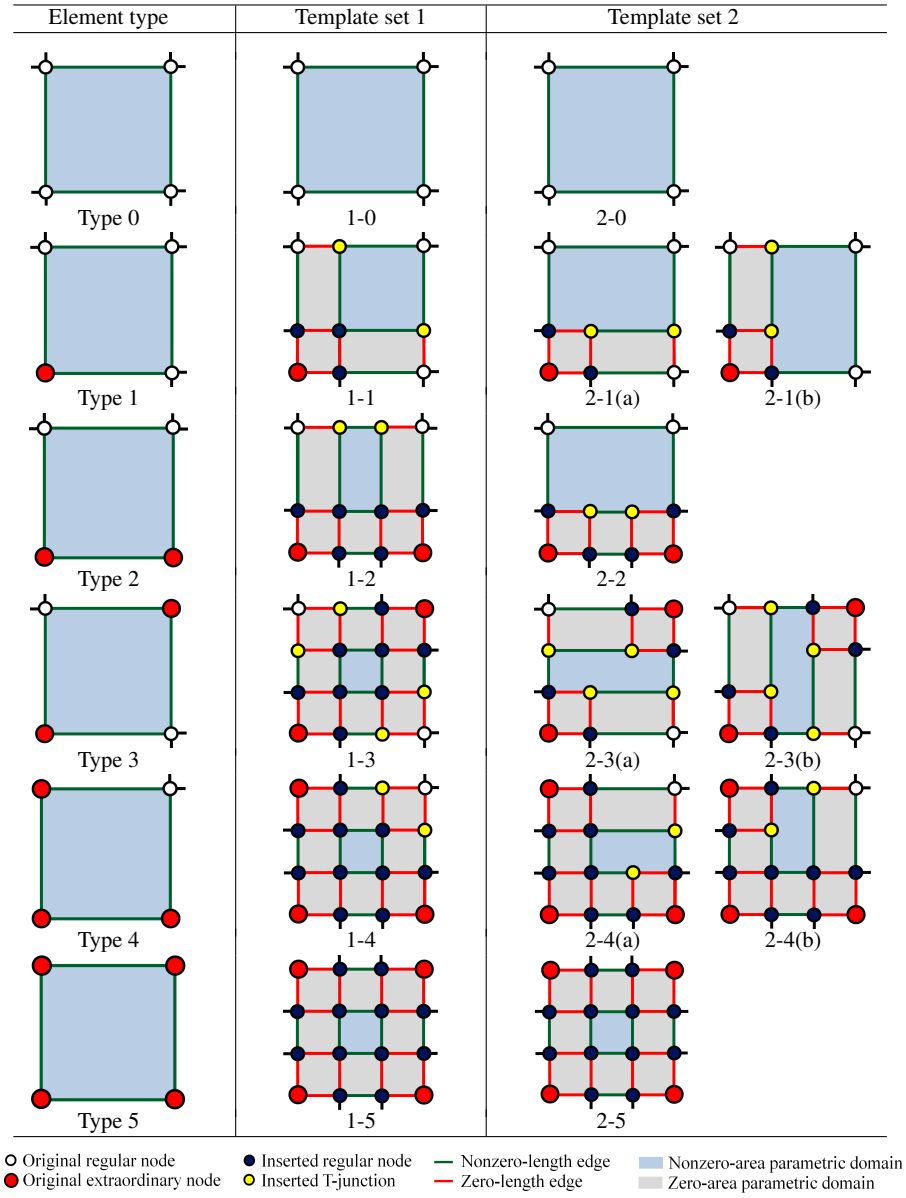
4.1.1 Two Sets of Templates

Template set 1 was derived using T-NURCC subdivision [13] and in this template set, there is a unique template for each element type. All the templates have a conforming 2-ring neighborhood, and they satisfy the requirements for a gap-free T-mesh around extraordinary nodes. This will be proved later; see Lemma 1 in Section 4.2. Figure 5(a) shows a typical example of a valence-3 extraordinary node O. Using this template set, the surface continuity is C^0 at the knot coordinate corresponding to this extraordinary point, and C^0 across the shared curve of the nonzero domain for each ring until the 4-ring. Therefore, the extraordinary point has a negative influence on the surface continuity until the 4-ring neighborhood. Figure 5(b) shows one local region of an input quadrilateral mesh and Figure 5(c) is a valid T-mesh after applying template set 1. In this input mesh, for each extraordinary node, there are no other extraordinary nodes in its 2-ring neighborhood, or there are at least three elements between each pair of extraordinary nodes. In this case, applying these templates automatically yields a valid T-mesh; see Lemma 4 in Section 4.3.

Template set 1 inserts a large number of nodes and zero-length edges, which decrease the local surface continuity and also have deleterious effects on basis continuity. Furthermore, newly inserted T-junctions are all on the boundary of the input quadrilateral elements, increasing the possibility that the generated T-junctions will interfere with each other, making the T-mesh semi-standard or non-standard. To minimize the number of newly inserted nodes and to decrease the size of the region influenced by the extraordinary nodes, we designed template set 2. All of these templates have a conforming 1-ring neighborhood, and they also satisfy the gap-free requirement, which will be proved in Section 4.2. Using this template set, the surface continuity is C^0 at the knot coordinate corresponding to the extraordinary point, C^0 across the shared curve of the nonzero area parametric domain for each ring until the 3-ring neighborhood, and C^1 outside the 3-ring neighborhood until the 4-ring neighborhood.

Discussion: Compared to template set 1, template set 2 not only inserts fewer nodes, but also results in better continuity around extraordinary nodes. Most importantly, it produces many fewer T-junctions on the element boundary, which is the boundary of a quadrilateral element in the input mesh, so it will be much easier to make the T-mesh standard. Figure 6 shows template set 2 configurations. Compared to Figure 5(a), Figures 6(a) and (b) contain fewer T-junctions on the element boundary and this decreases the possibility of interference with other extraordinary nodes. In addition, there are no neighboring zero-length edges on the 2-ring

Table 1. Two sets of templates for six quadrilateral element types.



neighborhood boundary, so the surface has better continuity around the extraordinary node and it is C^0-continuous across the shared curve of the nonzero area parametric domain for each ring until the 3-ring neighborhood, instead of the 4-ring. Another difference between these two template sets is: for template set 1, each extraordinary node has a conforming 2-ring neighborhood, while for template set 2, each extraordinary node has a conforming 1-ring neighborhood, which explains why template set 2 inserts fewer nodes.

4.1.2 Orientation Selection in Template Set 2

From Table 1, we can observe that for element types 0, 2 and 5, a unique template is designed. However, for element types 1, 3 and 4, there are two templates with different orientations. These are depicted in Figure 6(a) and (b). Therefore, the question arises: which orientation should be chosen for a certain case? Here is the idea we apply: Attempt to avoid generating a semi-standard or non-standard T-mesh when two or more extraordinary nodes interfere with each other. We utilize the following three procedures.

Procedure 1: Minimize the number of T-junctions. In template set 2 (Table 1), different orientations can be chosen for element types 1, 3 and 4. A semi-standard or non-standard T-mesh subset may be generated when two T-junctions interfere with each other, hence we always choose the orientation which gives fewer T-junctions. Figure 7(c) is a T-mesh with fewer T-junctions than the T-mesh in Figure 7(b).

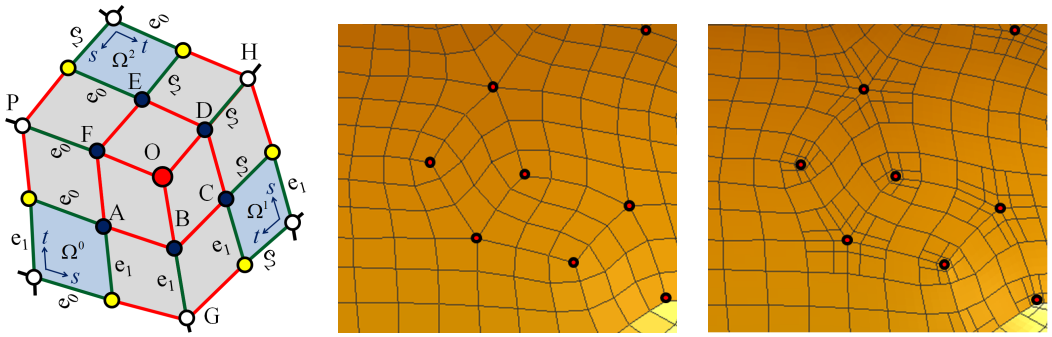


Fig. 5 Examples after applying template set 1. (a) A local region around a valence-3 extraordinary node O; (b) an input mesh with extraordinary nodes; and (c) the valid T-mesh. Red nodes are extraordinary nodes, and red edges have zero knot interval.

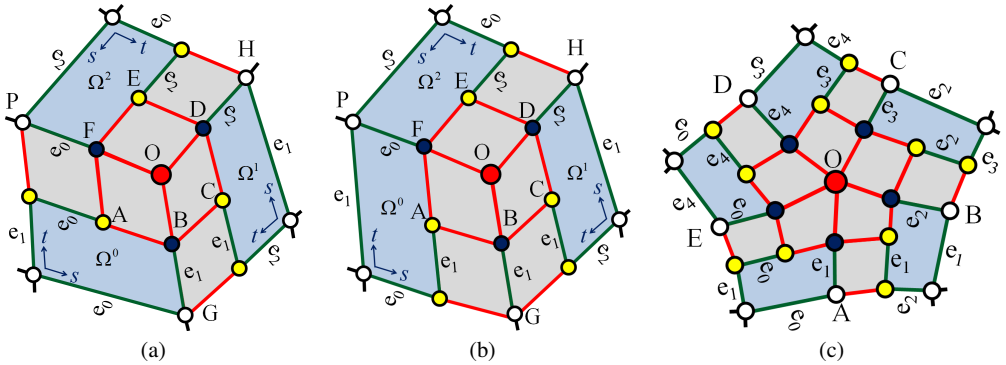


Fig. 6 Examples after applying template set 2. (a & b) Local regions around a valence-3 extraordinary node selecting different template orientations; and (c) a local region around a valence-5 extraordinary node. Red nodes are extraordinary nodes, and red edges have zero knot interval.

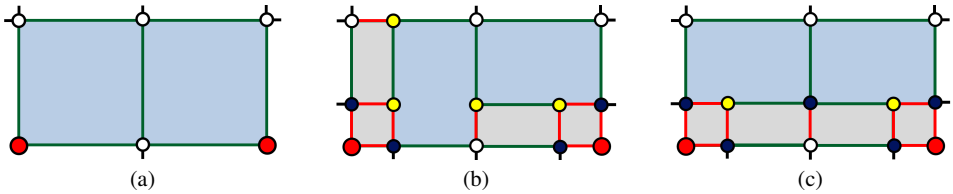


Fig. 7 Example for Procedure 1. (a) An input mesh; (b) one T-mesh with four T-junctions (inappropriate orientations are chosen); and (c) one T-mesh with two T-junctions (appropriate orientations are chosen, resulting in fewer T-junctions). Red nodes are extraordinary nodes, and yellow nodes are T-junctions.

Procedure 2: Attempt to insert T-junctions on the element boundary at the same parametric edges. In Figure 8, we chose the same orientation for the neighboring two elements and inserted T-junctions on the element boundary at the same parametric edges. In this way, we can avoid generating a semi-standard or non-standard T-mesh and facilitate the T-mesh standardization.

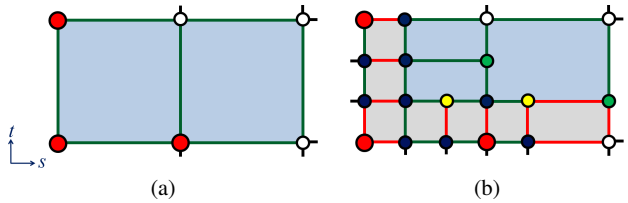
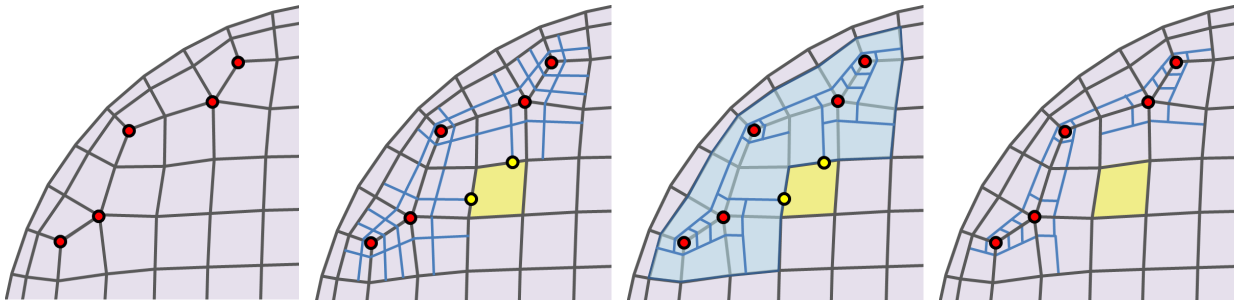


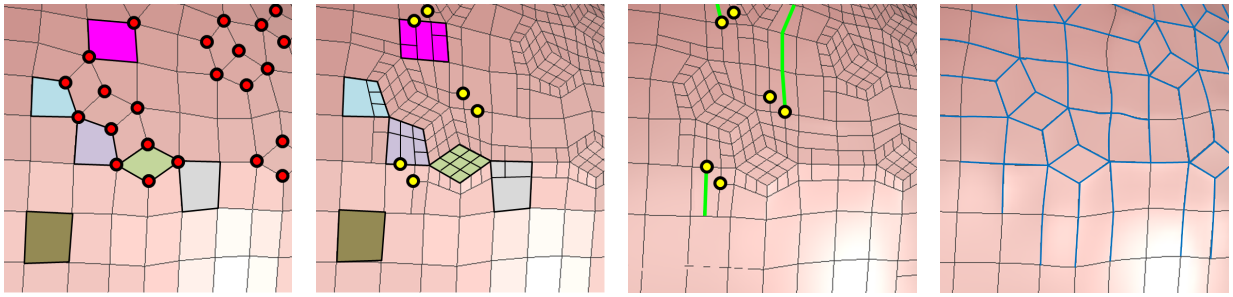
Fig. 8 Example for Procedure 2. (a) An input mesh; and (b) the T-mesh after inserting T-junctions on the element boundary at t-edges (the two green nodes). Red nodes are extraordinary nodes, and green and yellow nodes are T-junctions.

Procedure 3: Avoid inserting T-junctions in concave elements. Figure 9 shows an example. The red nodes are extraordinary nodes and the blue region is formed by all the quadrilateral elements with one or more extraordinary nodes. If this blue region is concave, we call the yellow element, which has two neighboring edges lying on the boundary of the blue region, a ***concave element***. As shown in Figure 9(b), template set 1 produces two T-junctions in the concave element and this makes the T-mesh non-standard. To make the T-mesh standard requires inserting many nodes and edges. For the same input mesh, using template set 2 with inappropriate orientations, as shown in Figure 9(c), results in the same problem. However, when we change the orientations



(a) (b) (c) (d)

Fig. 9 Example for Procedure 3. (a) An input mesh; (b) the generated T-mesh using template set 1; (c) the generated T-mesh using template set 2 with inappropriate orientations; and (d) a valid T-mesh after choosing appropriate orientations. Red points are extraordinary nodes, the blue region is formed by elements with extraordinary nodes and the yellow element is a concave element.



(a) (b) (c) (d)

Fig. 10 One local region with various element types. (a) An input mesh. Red nodes are extraordinary nodes, and one element of each type is identified by its color: tan (Type 0), grey (Type 1), blue (Type 2), magenta (Type 3), purple (Type 4) and green (Type 5); (b) the result after template application. The regions with yellow nodes are non-standard; (c) the valid T-mesh and T-spline surface after T-mesh standardization; (d) Bézier elements after surface fitting. The surface continuity across these blue lines is C^0 due to the inserted zero-length edges in (b) and (c).

and avoid inserting T-junctions in the concave element, we can obtain a standard T-mesh as shown in Figure 9(d). Comparing Figure 9(d) to Figure 9(b), we see the benefit of selecting appropriate orientation.

In our implementation, Procedure 1 has the highest priority and Procedure 3 has the lowest priority. Figure 10(a) shows a local region with various types of elements and Figure 10(b) is the result after applying template set 2. Note that there are still some semi-standard/non-standard local regions which cannot be avoided no matter which orientation is used, for example, the region around the yellow nodes. Thus, to obtain a valid T-mesh, the template application step needs to be followed by a T-mesh standardization step. Before proceeding to the standardization algorithm, we will first derive a sufficient condition to guarantee the generated T-spline surface is gap-free.

4.2 Sufficient Condition for Gap-free T-mesh around Extraordinary Nodes

To obtain a gap-free T-mesh, we need to determine a sufficient condition that can guarantee the local region around each extraordinary node is gap-free. This sufficient condition serves as a theoretical basis for our template development. Recall that the degree of the T-spline is 3.

Lemma 1 (Sufficient Condition for Gap-free T-mesh around Extraordinary Nodes) *If the T-mesh topology surrounding an extraordinary node is a conforming* 1*-ring neighborhood and all the knot intervals of the* 1*-ring neighborhood are zero, the local region around this extraordinary node is gap-free.*

Proof: As shown in Figure 11(a), the red node O is a valence-3 extraordinary node. The T-mesh topology around O is a conforming 1-ring neighborhood and all the knot intervals inside this 1-ring neighborhood are zero. Suppose the knot coordinates corresponding to O are located at the parametric origin. Ω^0, Ω^1 and Ω^2 are three nonzero area parametric domains around node O, and S^0, S^1 and S^2 are the T-spline patches defined by these three domains. In this paper, "domain" refers to one parametric area and "patch" refers to a T-spline surface defined on one domain. We first prove it is gap-free across the boundary shared by the two patches defined by Ω^0 and Ω^1, or the local region defined by $\Omega^0 \cup \Omega^1$ is gap-free. In other words, $S^0(0,t) = S^1(0,t)$, where $t \in [-e_1, 0]$.

In Ω^0 and Ω^1, all the nodes share the same knot vectors except the extraordinary node O. Hence, Ω^0 and Ω^1 share

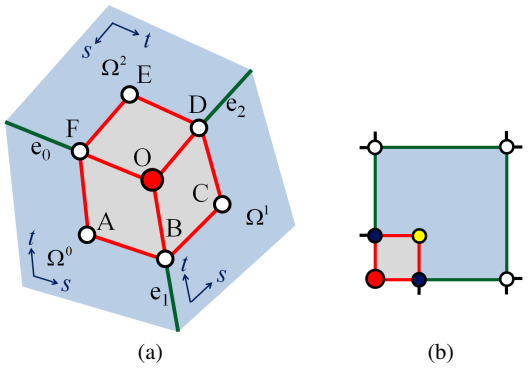


Fig. 11 Sufficient condition for a T-mesh around an extraordinary node to be gap-free. (a) A local region around a valence-3 extraordinary node (the red one); and (b) one L-shape face which is semi-standard. The red edges are zero-length edges. The knot coordinate corresponding to O is the parametric origin.

the same basis functions for all the nodes except O. The knot vectors of O are $\mathbf{s}_O^0 = [-e_0,0,0,0,0]$, $\mathbf{t}_O^0 = [-e_1,0,0,0,0]$ in Ω^0, and $\mathbf{s}_O^1 = [0,0,0,0,e_2]$, $\mathbf{t}_O^1 = [-e_1,0,0,0,0]$ in Ω^1. Based on the definition of a cubic B-spline, if there are knots of multiplicity 4 in the knot vector, the basis function value at the multiple knot is 1. Hence we have $N_O^s(0) = N_O^s[-e_0,0,0,0,0,0](0) = 1$ for Ω^0 and $N_O^s(0) = N_O^s[0,0,0,0,e_2](0) = 1$ for Ω^1. Then we can obtain

$$\begin{aligned} S^0(0,t) &= \frac{\sum_{i\neq O} w_i C_i N_i^s(0) N_i^t(t) + w_O C_O N_O^s(0) N_O^t(t)}{\sum_{i\neq O} w_i N_i^s(0) N_i^t(t) + w_O N_O^s(0) N_O^t(t)} \\ &= \frac{\sum_{i\neq O} w_i C_i N_i^s(0) N_i^t(t) + w_O C_O N_O^t(t)}{\sum_{i\neq O} w_i N_i^s(0) N_i^t(t) + w_O N_O^t(t)}, \end{aligned} \tag{5}$$

$$\begin{aligned} S^1(0,t) &= \frac{\sum_{i\neq O} w_i C_i N_i^s(0) N_i^t(t) + w_O C_O N_O^s(0) N_O^t(t)}{\sum_{i\neq O} w_i N_i^s(0) N_i^t(t) + w_O N_O^s(0) N_O^t(t)} \\ &= \frac{\sum_{i\neq O} w_i C_i N_i^s(0) N_i^t(t) + w_O C_O N_O^t(t)}{\sum_{i\neq O} w_i N_i^s(0) N_i^t(t) + w_O N_O^t(t)} \\ &= S^0(0,t). \end{aligned} \tag{6}$$

Therefore, the two patches defined by Ω^0 and Ω^1 share the same curve when $s = 0$ and $t \in [-e_1,0]$. In other words, the two patches are continuous or gap-free across the shared boundary. Due to symmetry, the surface is gap-free across the boundary shared by the two patches defined by Ω^1 and Ω^2 and this is also true for Ω^2 and Ω^0. Note that all the three patches share one corner point O. Hence the surface is gap-free for the 2-ring neighborhood. In addition, the basis function value corresponding to the extraordinary point is zero beyond the second ring, and all the other control points have unique basis functions. Since the B-Spline basis function is at least C^0-continuous, we can conclude that the surface is gap-free. □

Discussion: Lemma 1 provides one sufficient condition for a T-mesh to be gap-free around an extraordinary node. For each extraordinary node in the templates (Table 1), the 1-ring neighborhood is always conforming and all the edges of the 1-ring neighborhood have zero knot interval. In addition, since our goal is to generate standard T-meshes, the L-shape faces, for example the blue face in Figure 11(b), are not allowed in our templates because they will make the T-mesh semi-standard. Figure 5(a) shows one T-mesh obtained by applying template set 1. The 1-ring neighborhood for the extraordinary node O is obviously conforming and all the edges on the 1-ring neighborhood have zero knot interval. Similarly in Figure 6, the three T-meshes are obtained using template set 2 and it is obvious that no matter which orientations we choose, the 1-ring neighborhood for the extraordinary node is always conforming and their knot intervals are also zero. In summary, after applying our templates the T-meshes satisfy the gap-free requirement (Lemma 1) and the T-spline surfaces are gap-free.

4.3 Standard Configuration

To obtain a gap-free and standard T-mesh, we need to understand what T-mesh configurations are standard. Although the definition of a standard T-mesh is given in [14], characterizing standard T-mesh configurations is still an open problem. In our algorithm, the T-mesh obtained after applying the templates is quasi-uniform, which only contains edges with zero or unit knot interval. In this subsection, we will discuss what configuration yields a standard T-spline for a quasi-uniform T-mesh. Firstly, we define a new operation, called a type-preserving transformation.

Definition 4.4. A ***type-preserving transformation*** is an operation for a T-mesh in which nodes or edges are removed or inserted without changing the type of the T-spline.

In our algorithm, we found two type-preserving transformations for the quasi-uniform T-mesh obtained after applying the templates. The two type-preserving transformations can be used to remove T-junctions from the T-mesh and they are described in the following lemma.

Lemma 2 *In a quasi-uniform T-mesh, the following two node-removing transformations are type-preserving:*

1. *Three nodes A, B and C form two neighboring zero-length s-edges (or t-edges) and they share the same t (or s) knot vector (Figure 12(a-c)). If removing A or C does not change the knot vectors of their surrounding nodes (not including A, B and C), then this removal is a type-preserving transformation.*

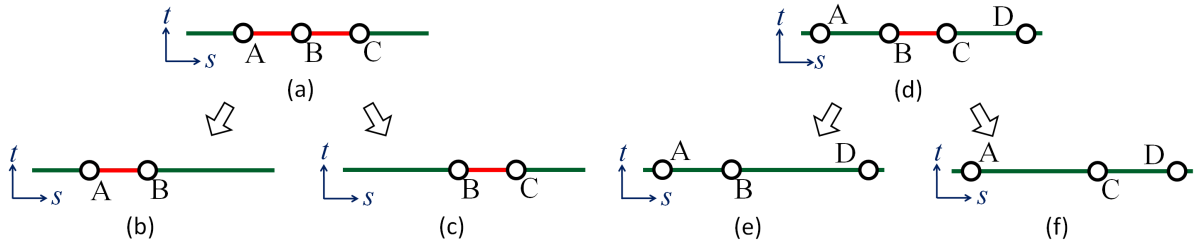


Fig. 12 Type-preserving transformations for quasi-uniform T-meshes. (a-c) Three nodes form two neighboring zero-length edges; and (d-f) two middle nodes of a set of four neighboring nodes form one zero-length edge. Red edges have zero knot interval. The knot coordinate corresponding to A is the parametric origin.

2. *Four neighboring nodes A, B, C and D share the same t (or s) knot vector and the two middle nodes B and C form one zero-length s-edge (or t-edge) (Figure 12(d-f)). If removing B or C does not change the knot vectors of their neighboring nodes (not including A, B, C and D), then this removal is a type-preserving transformation.*

Proof: In Figure 12(a-c), we suppose $\mathbf{s}_A = [s_0, s_1, 0, 0, 0]$, $\mathbf{s}_B = [s_1, 0, 0, 0, s_2]$ and $\mathbf{s}_C = [0, 0, 0, s_2, s_3]$, where $s_0 \le s_1 < 0$ and $0 < s_2 \le s_3$. Then if we remove node C, $\mathbf{s}_A$ and $\mathbf{s}_B$ become $[s_0, s_1, 0, 0, s_2]$ and $[s_1, 0, 0, s_2, s_3]$, respectively. Using the Oslo knot insertion algorithm [6], we can obtain

$$\begin{bmatrix} N[s_0,s_1,0,0,s_2](s) \\ N[s_1,0,0,s_2,s_3](s) \end{bmatrix} = T_0 \begin{bmatrix} N[s_0,s_1,0,0,0](s) \\ N[s_1,0,0,0,s_2](s) \\ N[0,0,0,s_2,s_3](s) \end{bmatrix}, \tag{7}$$

or

$$\begin{bmatrix} \overline{N_A^s} \\ \overline{N_B^s} \end{bmatrix} = T_0 \begin{bmatrix} N_A^s \\ N_B^s \\ N_C^s \end{bmatrix}, \tag{8}$$

where

$$T_0 = \begin{bmatrix} 1 & \frac{s_2}{s_2 - s_1} & 0 \\ 0 & \frac{-s_1}{s_2 - s_1} & 1 \end{bmatrix}. \tag{9}$$

Then we have

$$\overline{N_A^s} + \overline{N_B^s} = N_A^s + N_B^s + N_C^s. \tag{10}$$

Since $\mathbf{t}_A = \mathbf{t}_B = \mathbf{t}_C$ and removing C does not change $\mathbf{t}_A$ and $\mathbf{t}_B$, we have $N_A^t = N_B^t = N_C^t = N^t(t)$. Then we can obtain

$$\{\overline{N_A^s} + \overline{N_B^s}\} N^t(t) = \{N_A^s + N_B^s + N_C^s\} N^t(t). \tag{11}$$

In other words,

$$\sum_{i \in \{A,B\}} \overline{B}_i(s,t) = \sum_{i \in \{A,B,C\}} B_i(s,t). \tag{12}$$

In addition, this transformation does not change the knot vectors of the surrounding nodes (not including A, B and C). We can conclude that removing C does not change the summation of the basis functions in the T-mesh at any parametric position. In other words, this transformation is type-preserving. Since A is topologically symmetric with C, removing A is also a type-preserving transformation. Similarly, we can also prove that the reverse transformations from Figure 12(b) or Figure 12(c) to Figure 12(a), as well as the transformation from Figure 12(b) to Figure 12(c), are all type-preserving.

Similarly in Figure 12(d-f), we suppose $\mathbf{s}_A = [s_0, s_1, 0, s_2, s_2]$, $\mathbf{s}_B = [s_1, 0, s_2, s_2, s_3]$, $\mathbf{s}_C = [0, s_2, s_2, s_3, s_4]$, and $\mathbf{s}_D = [s_2, s_2, s_3, s_4, s_5]$, where $s_0 \le s_1 < 0$, $s_2 > 0$ and $s_2 \le s_3 \le s_4 \le s_5$. Then if we remove node C, $\mathbf{s}_A$, $\mathbf{s}_B$ and $\mathbf{s}_D$ become $[s_0, s_1, 0, s_2, s_3]$, $[s_1, 0, s_2, s_3, s_4]$ and $[s_0, s_2, s_3, s_4, s_5]$, respectively. Using the Oslo knot insertion algorithm [6], we obtain

$$\begin{bmatrix} N[s_0,s_1,0,s_2,s_3](s) \\ N[s_1,0,s_2,s_3,s_4](s) \\ N[0,s_2,s_3,s_4,s_5](s) \end{bmatrix} = T_1 \begin{bmatrix} N[s_0,s_1,0,s_2,s_2](s) \\ N[s_1,0,s_2,s_2,s_3](s) \\ N[0,s_2,s_2,s_3,s_4](s) \\ N[s_2,s_2,s_3,s_4,s_5](s) \end{bmatrix}, \tag{13}$$

or

$$\begin{bmatrix} \overline{N_A^s} \\ \overline{N_B^s} \\ \overline{N_D^s} \end{bmatrix} = T_1 \begin{bmatrix} N_A^s \\ N_B^s \\ N_C^s \\ N_D^s \end{bmatrix}, \tag{14}$$

where

$$T_1 = \begin{bmatrix} 1 & \frac{s_3 - s_2}{s_3 - s_1} & 0 & 0 \\ 0 & \frac{s_2 - s_1}{s_3 - s_1} & \frac{s_4 - s_2}{s_4} & 0 \\ 0 & 0 & \frac{s_2}{s_4} & 1 \end{bmatrix}. \tag{15}$$

Then we have

$$\overline{N_A^s} + \overline{N_B^s} + \overline{N_D^s} = N_A^s + N_B^s + N_C^s + N_D^s. \tag{16}$$

Since $\mathbf{t}_A = \mathbf{t}_B = \mathbf{t}_C = \mathbf{t}_D$ and removing C does not change $\mathbf{t}_A$, $\mathbf{t}_B$ and $\mathbf{t}_D$, we have $N_A^t = N_B^t = N_C^t = N_D^t = N^t(t)$. Then we can obtain

$$\{\overline{N_A^s} + \overline{N_B^s} + \overline{N_D^s}\} N^t(t) = \{N_A^s + N_B^s + N_C^s + N_D^s\} N^t(t). \tag{17}$$

In other words,

$$\sum_{i \in \{A,B,D\}} \overline{B}_i(s,t) = \sum_{i \in \{A,B,C,D\}} B_i(s,t). \tag{18}$$

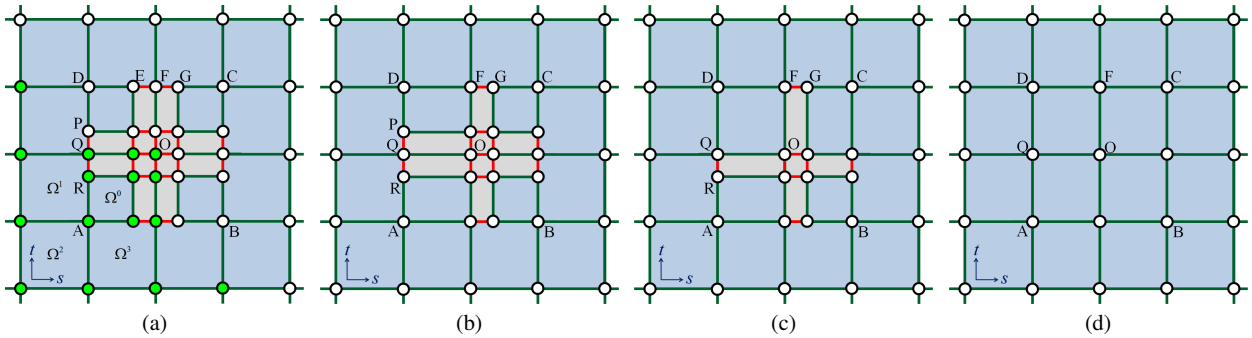


Fig. 13 Converting a complicated T-mesh configuration to a simple standard one through type-preserving transformations. (a) The original T-mesh; (b) one column is removed; (c) one row is removed; and (d) the resulting standard T-mesh after removing the second column and row. Green nodes have nonzero basis function values in Ω^0, Ω^1, Ω^2 and Ω^3.

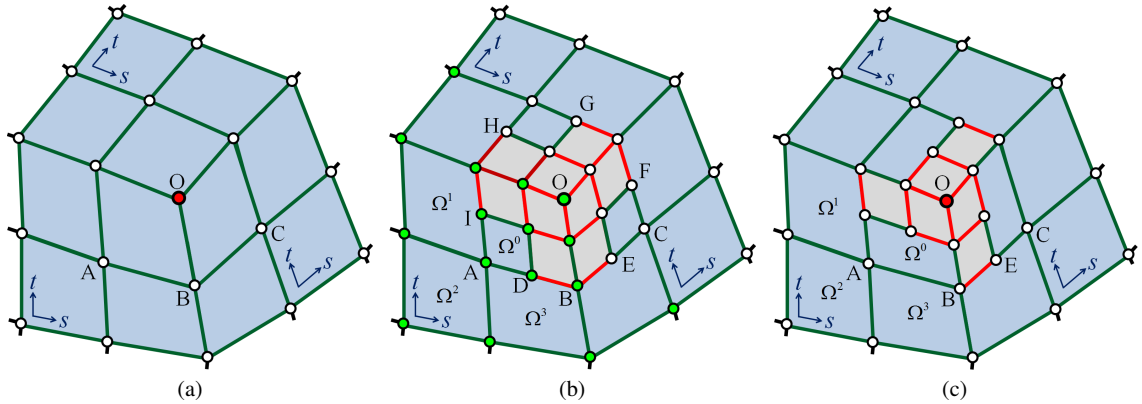


Fig. 14 One local region with one extraordinary node. (a) The input quadrilateral mesh; (b) the T-mesh after applying template set 1; and (c) the T-mesh after applying template set 2. Green nodes have nonzero basis function values in Ω^0, Ω^1, Ω^2 and Ω^3.

In addition, this transformation does not change the knot vectors of the surrounding nodes (not including A, B, C and D). We can conclude that removing C does not change the summation of the basis functions in the T-mesh at any parametric position, and the transformation is type-preserving. Since B is topologically symmetric with C, we can conclude that removing B is also type-preserving. □

There are three types of T-spline: standard, semi-standard and non-standard. In order to figure out the type of a given T-mesh without calculating $\sum_i B_i(s,t), \forall (s,t) \in \Omega$, we use type-preserving transformations to convert it to a simper configuration with a known T-spline type.

Lemma 3 (Sufficient Condition for Standard T-mesh Configuration) *A T-mesh is standard if it can be converted to a standard T-mesh through type-preserving transformations.*

Since a type-preserving transformation does not change the type of a T-mesh, it is obvious that the given T-mesh is also standard. Taking the T-mesh in Figure 13(a) as an example, we suppose there are no any extraordinary nodes on the 2-ring neighborhood of O in the input quadrilateral mesh. In this case, $\mathbf{t}_E = \mathbf{t}_F = \mathbf{t}_G$ and the three nodes E, F and G form two neighboring zero-length edges, hence we can use the first type-preserving transformation in Lemma 2 to remove node E. In the same way, the whole column with E can be removed as shown in Figure 13(b). Similarly, $\mathbf{s}_P = \mathbf{s}_Q = \mathbf{s}_R$ and the three nodes P, Q and R also form two neighboring zero-length edges. We can remove P and the whole row, see Figure 13(c). Figure 13(d) is obtained by removing the column with G and the row with R by way of the second type-preserving transformation in Lemma 2. In this way, we convert a complicated T-mesh configuration to a simple one by way of type-preserving transformations. It is obvious that the T-mesh in Figure 13(d) is standard, so we can conclude that the given T-mesh in Figure 13(a) is also standard. In the following, we will use this lemma and the two type-preserving transformations in Lemma 2 to study whether or not the T-mesh obtained after template development is standard.

Lemma 4 *For each extraordinary node in the input quadrilateral mesh, if there are no other extraordinary nodes in its 2-ring neighborhood, the cubic T-mesh obtained after applying template set 1 or template set 2 is standard.*

Proof: Figure 14(a) shows a local region around an extraordinary node O, and there are no other extraordinary nodes in

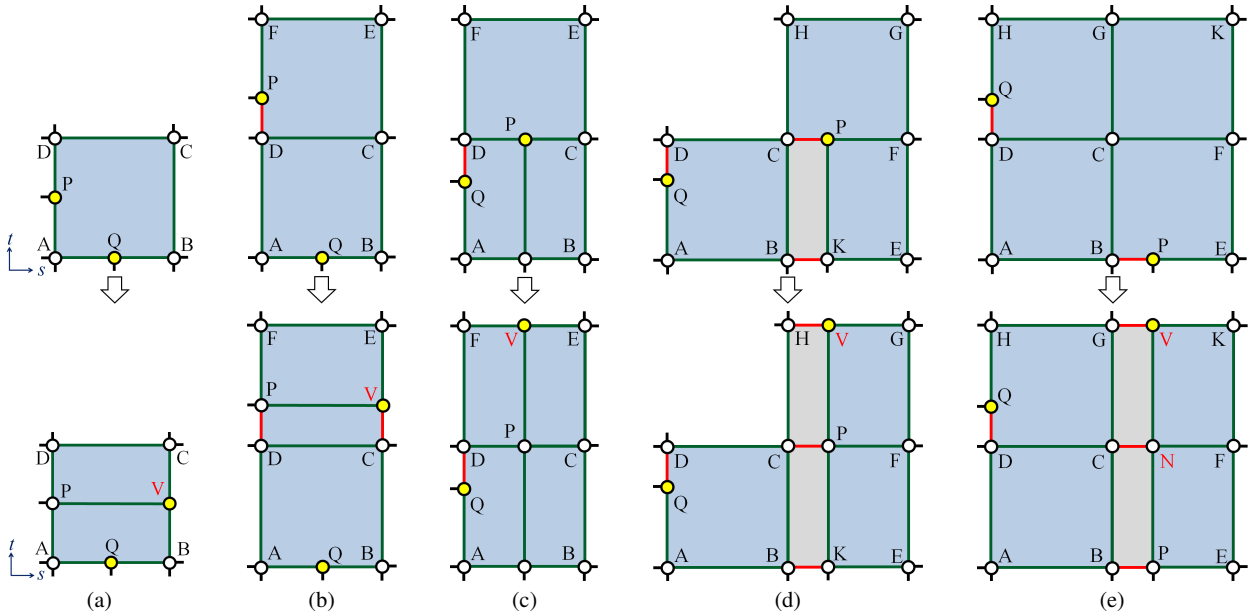


Fig. 15 Five semi-standard/non-standard configurations (top). New edges are inserted to make them standard (bottom). The yellow points are T-junctions.

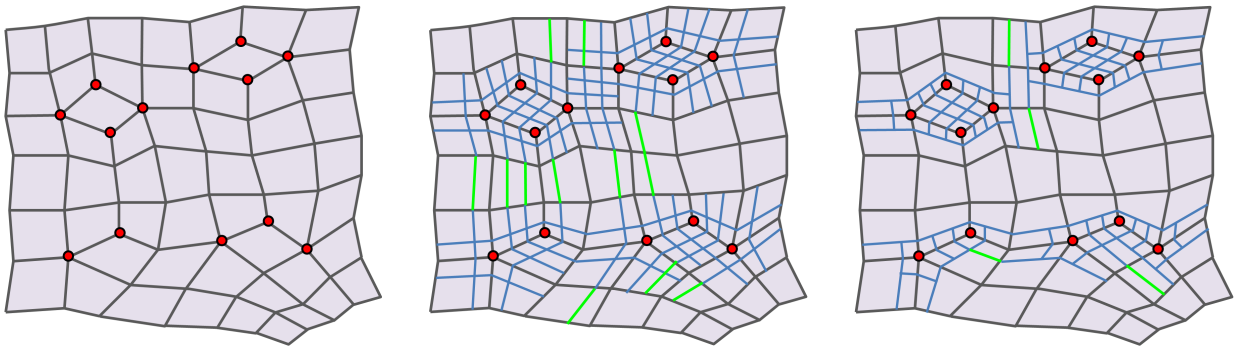
its 2-ring neighborhood. Figure 14(b) is the T-mesh after applying template set 1, and Figure 14(c) shows the result after applying template set 2. For the T-mesh in Figure 14(b), all the green nodes have nonzero basis function values in the four domains Ω^0, Ω^1, Ω^2 and Ω^3. We can observe that the basis functions of these green nodes are the same as the ones for the green nodes in Figure 13(a), which also have nonzero basis function values for Ω^0, Ω^1, Ω^2 and Ω^3. Since we have already proved the T-mesh in Figure 13(a) is standard, we can conclude that the T-mesh in Figure 14(b) is also standard for Ω^0, Ω^1, Ω^2 and Ω^3. Due to the symmetry, this subset of the T-mesh is standard for all the other domains. Furthermore, the T-junctions D, E, F, G, H and I can be removed by way of type-preserving transformations. Therefore, the extraordinary node O does not influence domains outside this subset, or its 3-ring neighborhood. For the T-mesh in Figure 14(c), since $\mathbf{t}_A = \mathbf{t}_B = \mathbf{t}_E$ we can insert one node D on the edge AB by the reverse of the first type-preserving transformation in Lemma 2. In the same way, nodes F and H can be inserted and then we convert this T-mesh configuration to the T-mesh in Figure 14(b), which has been proved to be standard. Therefore, the T-mesh in Figure 14(c) is standard as well. In summary, if no any other extraordinary nodes exist on the 2-ring neighborhood of one extraordinary node in the input mesh, the T-mesh after applying template set 1 or template set 2 is standard. □

Discussion: Here we only use one local region around a valence-3 extraordinary node to prove Lemma 4. For an extraordinary node with a different valence number, the topology of its one surrounding nonzero area parametric domain is the same. Hence, we can obtain the same conclusion for any extraordinary node. In general, if the input mesh contains adjacent extraordinary nodes, applying templates may not be able to yield a standard T-mesh directly and in this situation additional nodes need to be inserted as explained in Section 4.4.

4.4 T-mesh Standardization

During template development, generating a semi-standard or non-standard subset of a T-mesh is sometimes unavoidable. Hence, T-mesh standardization is a necessary step. In the obtained quasi-uniform T-mesh, we found five possible semi-standard/non-standard configurations (Figure 15) in which not all the T-junctions can be removed through type-preserving transformations. One important requirement for the two type-preserving transformations in Lemma 2 is that the related nodes must share the same knot vector along the s or t direction. However, in these five semi-standard or non-standard cases this requirement is not satisfied, therefore some T-junctions cannot be removed by type-preserving transformations. Can these T-mesh subsets be made standard? To standardize T-meshes we insert edges to make the related nodes have the same knot vector along one parametric direction.

For the example in Figure 15(a), one element contains two T-junctions, P and Q, lying on two different parametric edges. It is obvious that $\mathbf{t}_A \neq \mathbf{t}_Q$ and $\mathbf{s}_A \neq \mathbf{s}_P$. Hence, P and Q cannot be removed through type-preserving transformations. In order to make this subset standard, we insert one node V and one edge PV to make $\mathbf{t}_A = \mathbf{t}_Q = \mathbf{t}_B$ and also



(a) (b) (c)

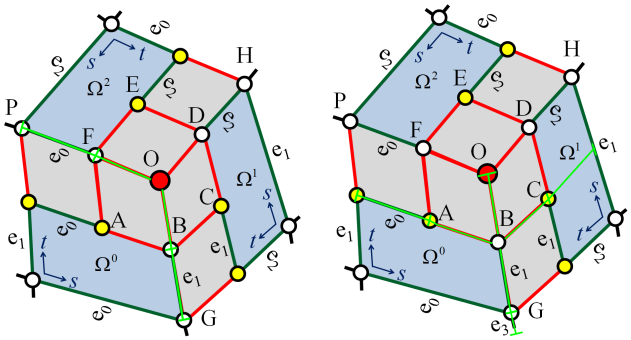
Fig. 16 Comparison of valid T-meshes generated using the two template sets. (a) An input quadrilateral mesh with extraordinary nodes (red ones); (b) the valid T-mesh using template set 1; and (c) the valid T-mesh using template set 2.

make one T-junction regular. After that, if AQ has zero knot interval, then this subset is already standard, since Q can be removed using a type-preserving transformation and then V can be removed in the same way. Otherwise, if QB has zero knot interval, the subset has the same topology as in Figure 15(c), which can also be standardized by inserting nodes and edges. Similarly in Figure 15(b-e), the two T-junctions P and Q cannot be removed through type-preserving transformations. Therefore, we insert nodes to convert the T-junction P to a regular node. During the implementation, we detect these cases and insert nodes iteratively until there are no such semi-standard/non-standard local regions. In this way, the T-mesh is standardized.

Figure 16 shows two valid T-meshes generated using the two template sets. Comparing the two valid T-meshes, we conclude that template set 2 generates many fewer nodes and edges. The blue edges are generated in the template development step, and the green edges are inserted during T-mesh standardization. Figure 10(c) shows another valid T-mesh after T-mesh standardization.

4.5 Knot Vector Inference

After we obtain a valid T-mesh, the local knot vectors for each node need to be inferred from the T-mesh. For a regular node or a T-junction, its knot vectors can be inferred using the knot vector inference rule given in Section 2. For an extraordinary node and its adjacent nodes, the knot vectors should be inferred using different procedures. In Figure 17(a), node O corresponds to the parametric origin, that is, its knot coordinates are $(0,0)$. For domain Ω_0, the two left knots of $\mathbf{s}_O^0$ are obtained in the same way as a regular node, by sending out a ray along the negative s direction. The two right knots of $\mathbf{s}_O^0$ repeat O's s knot coordinate. Then we have $\mathbf{s}_O^0 = [-e_0, 0, 0, 0, 0]$, and $\mathbf{t}_O^0$ can be found in the same way, $\mathbf{t}_O^0 = [-e_1, 0, 0, 0, 0]$. Similarly, we have $\mathbf{s}_O^1 = [-e_1, 0, 0, 0, 0]$ and $\mathbf{t}_O^1 = [-e_2, 0, 0, 0, 0]$ for domain Ω_1. For nodes adjacent to an extraordinary node, the rule is similar to the knot vector inference rule for regular nodes and T-junctions, except that when the ray intersects the extraordinary node, the knot corresponding to this extraordinary node is repeated. For example, in Figure 17(b), O corresponds to the parametric origin, and we have $\mathbf{s}_B^0 = [-e_0, 0, 0, 0, e_2]$ and $\mathbf{t}_B^0 = [-e_3 - e_1, -e_1, 0, 0, 0]$ in Ω_0.

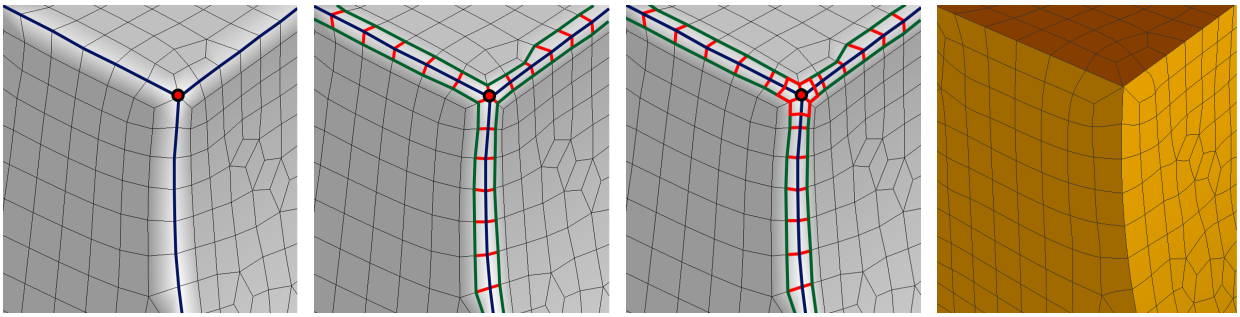


(a) (b)

Fig. 17 Knot vector inference for an extraordinary node and its adjacent node in the T-mesh. (a) Knot vector inference for an extraordinary node O; and (b) knot vector inference for a node B adjacent to the extraordinary node O. The knot coordinate corresponding to O is the parametric origin.

5 Sharp Feature Preservation

To maintain sharp features present in the input geometry, we insert zero-length edges across sharp edges and around sharp corners in the T-mesh. We duplicate sharp edges for each face and all the transverse edges have zero knot interval. Since all the sharp edges are shared by two faces, there will be two neighboring zero-length edges transverse to each sharp edge. Therefore, as shown in Figure 18, we obtain three repeating knots across the sharp edges and the continuity across sharp edges is C^0. For each sharp corner, we duplicate all the element edges connecting to this corner for each incident sharp edge, and all the edges of its 1-ring neighborhood have zero knot interval. In this way, all sharp features are preserved.



(a) (b) (c) (d)

Fig. 18 Sharp feature preservation. (a) An input quadrilateral mesh with sharp edges (blue edges) and a sharp corner (the red point); (b) the sharp edges are duplicated (green edges) and zero-length edges (red edges) are inserted across them; (c) zero-length edges (red edges) are inserted around the sharp corner; and (d) Bézier elements are extracted from the constructed T-spline surface.

6 Surface Fitting and Continuity Improvement

From the topology stage, we obtain a standard and gap-free T-mesh. In the geometry stage, we relocate control points to improve the surface accuracy and continuity.

6.1 Surface Fitting

We relocate control nodes in the T-mesh so that the output T-spline surface will interpolate all the nodes in the input quadrilateral mesh. The constructed T-spline will satisfy

$$S(s_j,t_j)=\frac{\sum_{i=0}^{n} w_i C_i B_i(s_j,t_j)}{\sum_{i=0}^{n} w_i B_i(s_j,t_j)}=P_j, \tag{19}$$

where C_i is a control point in the T-mesh, P_j is a vertex in the input quadrilateral mesh, and (s_j,t_j) are the parametric coordinates corresponding to P_j. A straightforward way to satisfy Equation (19) is to consider C_i as unknown and build a linear system of size n, where n is the number of control points. This method is very time-consuming for a T-mesh with many control points.

An alternative way to satisfy Equation (19) is to take advantage of the locality of T-splines. For each nonzero domain of the T-spline (for example, Ω^0 in Figure 19), we take the positions of the four nodes (P,G,O and Q), which come from the input mesh and whose parametric coordinates correspond to the four corners of the domain, as unknown variables and fix all the other nodes with nonzero basis function values. Suppose V_P, V_G, V_O, V_Q are the corresponding four vertices in the input quadrilateral mesh, and S^0 is the patch defined by domain Ω^0. Note that we have three sets of points: the vertices in the input mesh (for example, V_O), the T-mesh control points (for example, node O), and the points on the T-spline surface (for example, points on S^0). Then we make the defined T-spline patch S^0 interpolate these four

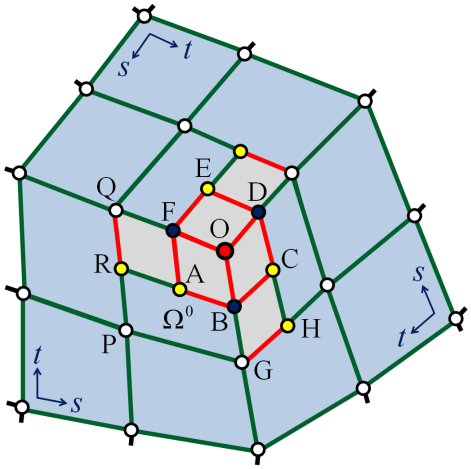
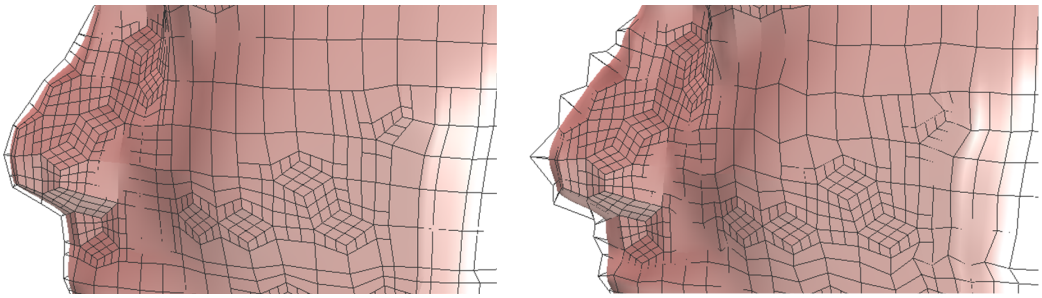


Fig. 19 The surface fitting scheme to make the constructed T-spline interpolate all the nodes in the input quadrilateral mesh.

vertices in the input mesh at the four corners. In this way we have four unknown variables and four equations. We solve a small linear system and obtain the new positions of the four nodes, $P^{'},G^{'},O^{'}$ and $Q^{'}$. These four nodes are then moved toward their new positions. In addition, we update all the nodes in the T-mesh with the same parametric coordinates as these four nodes. For instance, nodes A, B, C, D, E and F are relocated along the direction of $OO^{'}$, since they share the same parametric coordinates (0, 0) with O. We loop over each nonzero domain and iterate until the maximum distance between the four vertices of each quadrilateral element and the four corners of the corresponding T-spline patch is within a given error tolerance. Figure 20 shows an example of surface fitting in which the surface in Figure 20(b) interpolates all the nodes in the input mesh.

For a CAD model, if its NURBS representation is given and all the nodes in the input quadrilateral mesh are on the NURBS surface, we define metrics to measure the error between the constructed T-spline surface and the NURBS surface. Suppose I_N and Ω_N are NURBS parametric domains, and I and Ω are T-spline parametric domains. $C_N(u)$ de-



(a) (b)

Fig. 20 One surface fitting result. (a) A valid T-mesh with its defined T-spline surface before surface fitting; and (b) the T-mesh with its defined T-spline surface after surface fitting.

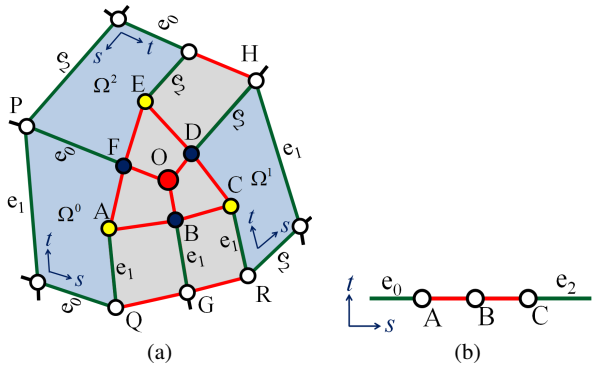
notes a NURBS curve defined on I_N, and $S_N(u,v)$ denotes a NURBS surface defined on Ω_N. Similary, $C(s)$ denotes a curve on a T-spline surface defined in I, and $S(s,t)$ denotes a T-spline surface defined on Ω. The error of one sharp edge is defined as

$$\varepsilon_c = \max_{s\in I}\left[\frac{1}{\lambda}\min_{u\in I_N} \| C(s) - C_N(u) \|\right], \tag{20}$$

and the surface error between the constructed T-spline and the given NURBS is defined as

$$\varepsilon_s = \max_{(s,t)\in\Omega}\left[\frac{1}{\lambda}\min_{(u,v)\in\Omega_N} \| S(s,t) - S_N(u,v) \|\right], \tag{21}$$

where $\lambda = (\sum_{i=1}^{3} \| B_{i+1} - B_i \| + \| B_1 - B_4 \|)/4$, and B_i ($i = 1,2,3$ and 4) are the four corners of the T-spline element containing $C(s)$ or $S(s,t)$. After surface fitting, the resulting T-spline interpolates all the nodes in the input mesh.



(a) (b)

Fig. 21 Continuity improvement via adjusting control nodes. (a) A local region around an extraordinary node (the red one); and (b) two neighboring zero-length edges. The knot coordinate corresponding to O is the parametric origin.

6.2 Continuity Improvement

The T-spline surface is not C^2-continuous around extraordinary nodes. There are two ways to improve continuity. One is to narrow the lower order continuity domain by local refinement. This method can improve continuity and, in the limit, make the surface C^2-continuous almost everywhere. However, it requires many additional nodes, especially when the model has a large number of extraordinary nodes. The other way to improve continuity is by adjusting control point locations. In our algorithm, we insert zero-length edges in the T-mesh, reducing surface continuity. Continuity can be improved by adjusting control point locations on these zero-length edges. For two neighboring patches, the continuity across their shared boundary depends on whether or not they share the same surface normal at the boundary. For example in Figure 21(a), the two patches $S^0(s,t)$ and $S^1(s,t)$ are defined by Ω^0 and Ω^1, respectively. They are represented in a homogeneous form,

$$S^0(s,t) = \sum_{i=0}^{n} \overline{C}_i N_i^s(s) N_i^t(t), \quad (s,t)\in\Omega^0, \tag{22}$$

$$S^1(s,t) = \sum_{i=0}^{n} \overline{C}_i N_i^s(s) N_i^t(t), \quad (s,t)\in\Omega^1, \tag{23}$$

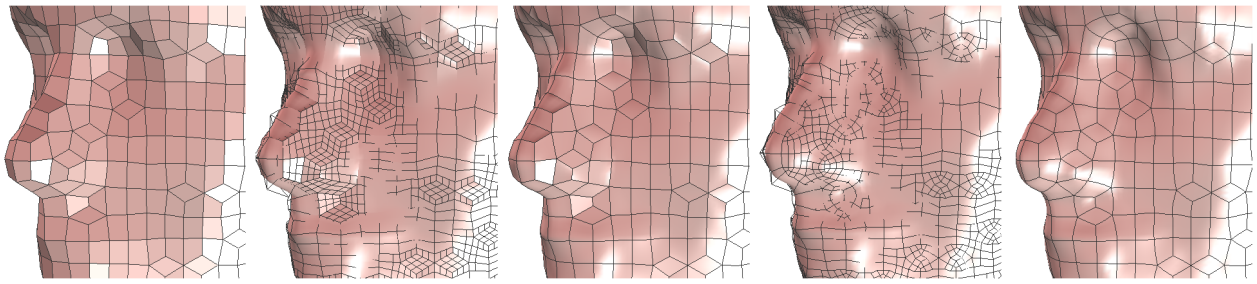
where $\overline{C}_i = [w_i x_i, w_i y_i, w_i z_i, w_i]$ is a control point in the homogeneous representation. Suppose $S^0(s,t)$ and $S^1(s,t)$ share the same curve when $s = 0$ and $t \in [-e_1, 0]$ (node O corresponds to the parametric origin). We can derive the surface normal of $S^0(s,t)$ at the common curve,

$$\begin{aligned}\mathbf{n}^0(t) &= \frac{\partial S^0}{\partial s}(0^-,t) \times \frac{\partial S^0}{\partial t}(0^-,t) \\ &= \sum_{i=0}^{n}\left[\overline{C}_i \frac{\partial N_i^s(s)}{\partial s}(0^-) N_i^t(t) \times \frac{\partial S^0}{\partial t}(0^-,t)\right],\end{aligned} \tag{24}$$

where $\times$ denotes the cross product. Similarly, we can get the normal of $S^1(s,t)$,

$$\begin{aligned}\mathbf{n}^1(t) &= \frac{\partial S^1}{\partial s}(0^+,t) \times \frac{\partial S^1}{\partial t}(0^+,t) \\ &= \sum_{i=0}^{n}\left[\overline{C}_i \frac{\partial N_i^s(s)}{\partial s}(0^+) N_i^t(t) \times \frac{\partial S^1}{\partial t}(0^+,t)\right].\end{aligned} \tag{25}$$

Since $S^0(s,t)$ and $S^1(s,t)$ share the same curve, $\frac{\partial S^0}{\partial t}(0^-,t) = \frac{\partial S^1}{\partial t}(0^+,t)$. If all the control points with nonzero basis



(a) (b) (c) (d) (e)

Fig. 22 Continuity improvement result via adjusting control nodes. (a) The input mesh; (b) the T-mesh with its defined T-spline surface before continuity improvement; (c) Bézier elements extracted from the T-spline in (b); (d) the T-mesh with its defined T-spline surface after continuity improvement; and (e) Bézier elements extracted from the T-spline in (d).

function values in domain Ω^0 and Ω^1 satisfy

$$\frac{\partial N_i^s(s)}{\partial s}(0^-)N_i^t(t) = \frac{\partial N_i^s(s)}{\partial s}(0^+)N_i^t(t), \tag{26}$$

we have

$$\mathbf{n}^0(t) = \mathbf{n}^1(t). \tag{27}$$

In other words, the surface is at least C^1-continuous across the shared curve. As shown in Figure 21, when there are two adjacent zero-length s or t edges, the control points A, B and C do not satisfy Equation (26), and the surface continuity is only C^0. However, if $\mathbf{t}_A = \mathbf{t}_B = \mathbf{t}_C$, we have $N_A^t(t) = N_B^t(t) = N_C^t(t)$. In addition, using the definition of a B-spline, we get $\frac{\partial N_A^s(s)}{\partial s}(0^-) = -\frac{3}{e_0}$, $\frac{\partial N_B^s(s)}{\partial s}(0^-) = \frac{3}{e_0}$ and $\frac{\partial N_C^s(s)}{\partial s}(0^-) = 0$. Thus, we have

$$\begin{aligned}
&\sum_{i\in\{A,B,C\}} \overline{C_i}\frac{\partial N_i^s(s)}{\partial s}(0^-)N_i^t(t) \times \frac{\partial S^0}{\partial t}(0^-,t) \\
&= \overline{C_A}(-\tfrac{3}{e_0})N_A^t(t) \times \tfrac{\partial S^0}{\partial t}(0^-,t) + \overline{C_B}(\tfrac{3}{e_0})N_B^t(t) \times \tfrac{\partial S^0}{\partial t}(0^-,t) \\
&= (\overline{C_B} - \overline{C_A})(\tfrac{3}{e_0})N_B^t(t) \times \tfrac{\partial S^0}{\partial t}(0^-,t).
\end{aligned} \tag{28}$$

Similarly,

$$\begin{aligned}
&\sum_{i\in\{A,B,C\}} \overline{C_i}\frac{\partial N_i^s(s)}{\partial s}(0^+)N_i^t(t) \times \frac{\partial S^0}{\partial t}(0^+,t) \\
&= \overline{C_B}(-\tfrac{3}{e_2})N_B^t(t) \times \tfrac{\partial S^0}{\partial t}(0^+,t) + \overline{C_C}(\tfrac{3}{e_2})N_C^t(t) \times \tfrac{\partial S^0}{\partial t}(0^+,t) \\
&= (\overline{C_C} - \overline{C_B})(\tfrac{3}{e_2})N_B^t(t) \times \tfrac{\partial S^0}{\partial t}(0^+,t).
\end{aligned} \tag{29}$$

If we set $(\overline{C_B} - \overline{C_A})(\frac{3}{e_0}) = (\overline{C_C} - \overline{C_B})(\frac{3}{e_2})$, we obtain

$$\begin{aligned}
&\sum_{i\in\{A,B,C\}} \overline{C_i}\frac{\partial N_i^s(s)}{\partial s}(0^-)N_i^t(t) \times \frac{\partial S^0}{\partial t}(0^-,t) \\
&= \sum_{i\in\{A,B,C\}} \overline{C_i}\frac{\partial N_i^s(s)}{\partial s}(0^+)N_i^t(t) \times \frac{\partial S^0}{\partial t}(0^+,t).
\end{aligned} \tag{30}$$

In other words, if A, B and C are collinear and $\frac{|AB|}{|BC|} = \frac{e_2}{e_0}$, then the two neighboring zero-length edges will not decrease the surface continuity.

Remark: If the knot vectors of the control points with nonzero basis function values in Ω^0 or Ω^1 do not have repeating knots at $s = 0$, $S^0(s,t)$ and $S^1(s,t)$ are C^2-continuous. For the two neighboring zero-length edges in Figure 21(b), we have $e_0 = e_2 = 1$ because the T-mesh is quasi-uniform. To satisfy Equation (30), we move node B to the middle between nodes A and C. Then $\frac{|AB|}{|BC|} = \frac{e_2}{e_0} = 1$. For the region around an extraordinary node, as shown in Figure 21(a), we move all the green nodes to the middle between two neighboring yellow nodes. The surface is C^0-continuous across the shared curve in the 2-ring, C^1-continuous from the 3-ring boundary to the 4-ring boundary, and C^2-continuous beyond the 4-ring neighborhood.

As shown in Figure 22, it is obvious that the surface continuity for the T-spline in Figures 22(d) and (e) is improved compared with the T-spline before continuity improvement, especially for the local region around each extraordinary node. We can also observe the geometry continuity improvement compared with the input mesh. However, this method decreases geometric accuracy and the constructed T-spline surface may not interpolate vertices in the input mesh. In addition, this method will not improve the continuity of the space formed by the basis functions in general.

7 Bézier Extraction from T-splines

Bézier elements are extracted from the constructed T-spline surface to facilitate isogeometric analysis [5, 12]. For a quasi-uniform T-mesh, each nonzero area parametric domain defines one Bézier element. The number of Bézier elements in the output T-spline is the same as the number of quadrilateral elements in the input mesh. For each nonzero area parametric domain, we determine the nodes with nonzero basis function values and then calculate the transformation matrix M^e between the T-spline basis functions and the Bézier basis

Table 2. Statistics of all the tested models

	Input mesh (vertex#, element#)	Number of extraordinary nodes	Number of T-mesh nodes	Time (s)	Number of Bézier elements	Error
Head	(2912, 2912)	1227	14519	1.015	2912	-
Knee	(1357, 1354)	685	7745	0.422	1354	-
Venus	(1554, 1552)	690	8111	0.500	1552	-
Multiaxis	(1478, 1476)	84	2600	0.500	1476	5.0e-3
Hook	(2244, 2242)	558	8131	0.781	2242	6.1e-3
Hook2	(2592, 2590)	494	8018	0.875	2590	3.1e-3
Varco	(4154, 4152)	515	10182	1.422	4152	2.3e-3

functions. We have

$$B_t^e = M^e B_b^e, \tag{31}$$

where

$$B_t^e = \left[B_0^e \; B_1^e \; B_2^e \; \cdots \; B_{n^e-2}^e \; B_{n^e-1}^e \right]^T, \tag{32}$$

and

$$B_b^e = \begin{bmatrix} N[0,0,0,0,1](s)N[0,0,0,0,1](t) \\ N[0,0,0,1,1](s)N[0,0,0,0,1](t) \\ N[0,0,1,1,1](s)N[0,0,0,0,1](t) \\ \vdots \\ N[0,0,1,1,1](s)N[0,1,1,1,1](t) \\ N[0,1,1,1,1](s)N[0,1,1,1,1](t) \end{bmatrix}. \tag{33}$$

Here, B_t^e is the vector formed by the T-spline basis functions with nonzero function values, B_b^e is the vector formed by the Bézier basis functions, and n^e is the number of nodes with nonzero basis function values in this domain. M^e can be calculated using the Oslo knot insertion algorithm [6]. Figure 10(d) shows the Bézier elements extracted from the constructed T-spline surface. Each pair of Bézier elements sharing blue edges is C^0-continuous across their common edge, due to the inserted zero-length edges in Figure 10(b) and (c). Figures 18(d) and 22(c, e) show more results of Bézier element extraction.

8 Numerical Examples and Discussion

We have applied the algorithm to several human anatomy models (Figures 1, 23 and 24) and CAD models with sharp features (Figures 25-28). As shown in Table 2, there are a large percentage of extraordinary nodes in each input quadrilateral mesh. The more extraordinary nodes in the input mesh, the more new nodes need to be inserted. The algorithm is robust and efficient. All the results were computed on a PC equipped with Intel Q6600 (4 cores, 2.4GHz) processor and 4GB main memory (DDR2, 800MHz).

In all the tested models, template set 2 is chosen because it introduces fewer nodes and zero-length edges. From Figures 1 and 23-28, we can observe that the number of Bézier elements is the same as the number of quadrilateral elements in the input mesh, despite the fact that the input quadrilateral and Bézier elements are different. See the caption of Figure 1 for the explanation. The algorithm works for complicated geometries, and we do not need to take into account topology, or check genus. We only need to check the valence number of each node to identify which nodes are extraordinary nodes. Sharp features are very common in CAD models, and here we applied the algorithm to four CAD models: multi-axis, hook, hook2 and varco. It is apparent that the T-spline surface preserves all the sharp features of the input model. For the CAD models, the error between the constructed T-spline and the input NURBS surface, given by Equations (20) and (21), is presented in Table 2.

9 Conclusions

We have developed a robust and efficient two-stage algorithm to convert any unstructured quadrilateral mesh to a standard T-spline surface. In the topology stage, we take the input mesh as the initial T-mesh and then make the initial T-mesh gap-free and standard. Two sufficient conditions are derived to guarantee the generated T-spline is gap-free around extraordinary nodes, and to decide which T-mesh configurations yield a standard T-spline. These two conditions provide a theoretical basis for template development and T-mesh standardization. In the geometry stage, an efficient surface fitting technique is developed to make the output T-spline interpolate all the nodes in the input mesh. The surface continuity is improved by adjusting control points around extraordinary nodes. In addition, the conversion algorithm preserves sharp features present in the input model. The output T-spline models can be used directly for isogeometric analysis. We intend to focus on improvements and simplifications in our future work, and extension to hexahedral meshes.

Acknowledgements We would like to thank Jin Qian for providing the quadrilateral meshes and Deborah Grunner for proofreading the paper. The work of W. Wang and Y. Zhang was supported in part by ONR Grant N00014-08-1-0653. The work of M.A. Scott and T.J.R. Hughes was supported by ONR Grant N00014-08-1-0992 and a grant from SINTEF. M.A. Scott was also partially supported by an ICES Graduate fellowship, and T.J.R Hughes was partially supported by NSF GOALI CMI-0700807/0700204.

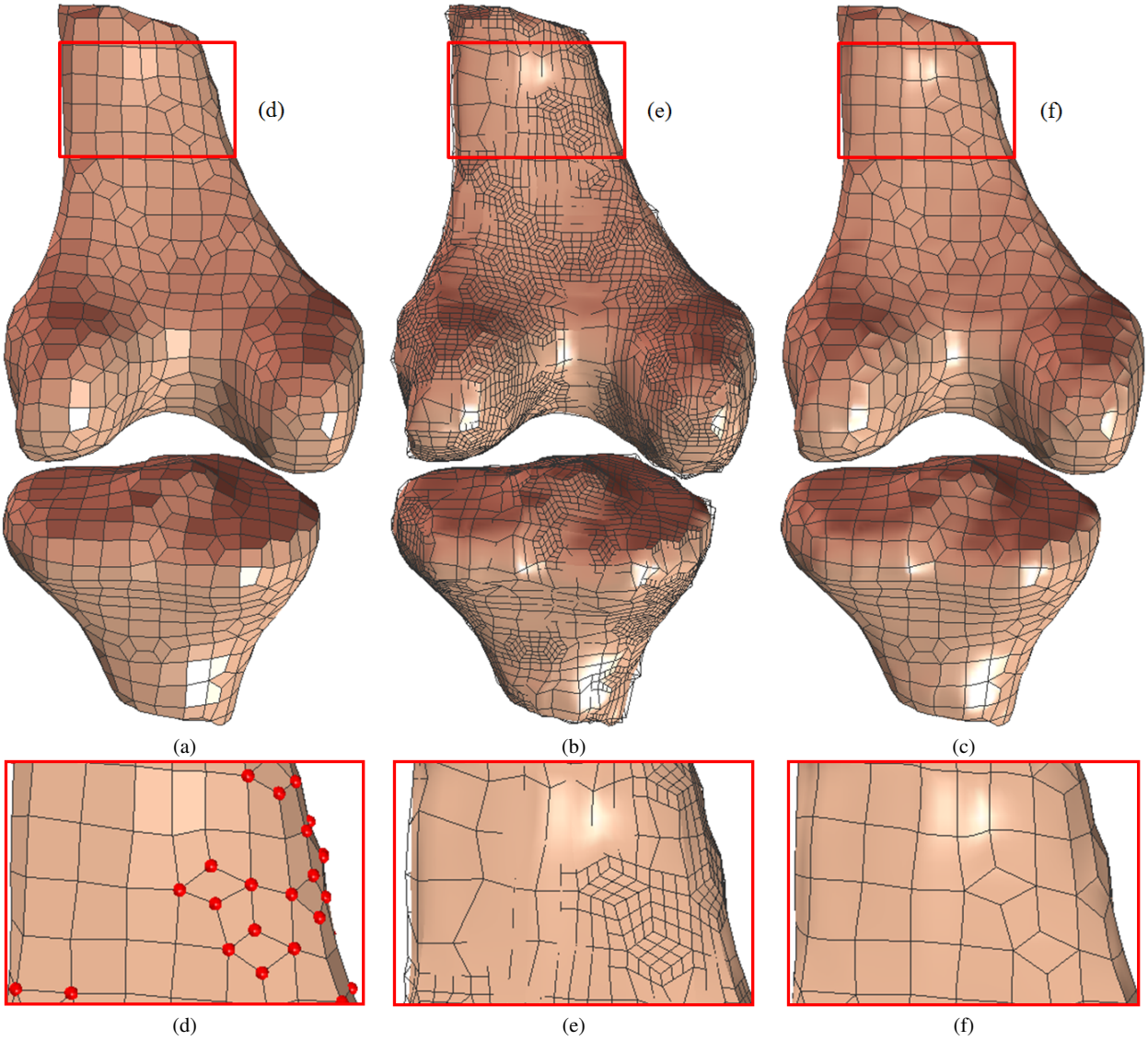


Fig. 23 The knee model. (a) The input unstructured quadrilateral mesh; (b) the constructed T-spline surface and T-mesh; and (c) the extracted Bézier elements. (d) to (f) show details, and red points in (d) denote extraordinary points.

References

1. C. W. Anderson and S. Crawford-Hines. Fast generation of NURBS surfaces from polygonal mesh models of human anatomy. Technical report, Colorado State University, 2000.
2. Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):229–263, 2010.
3. D.J. Benson, Y. Bazilevs, M.C. Hsu, and T.J.R. Hughes. Isogeometric shell analysis: The Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):276–289, 2010.
4. D.J. Benson, Y. Bazilevs, E. De Luycker, M.C. Hsu, M.A. Scott, T.J.R. Hughes, and T. Belytschko. A generalized finite element formulation for arbitrary basis functions: From isogeometric analysis to XFEM. *International Journal for Numerical Methods in Engineering*, 83(6):765–785, 2010.
5. M.J. Borden, M.A. Scott, J.A. Evans, and T.J.R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering, accepted*, 2010.
6. R. Goldman and T. Lyche. *Knot insertion and deletion algorithms for B-spline curves and surfaces*. Society for Industrial and Applied Mathematics–Philadelphia, 1993.

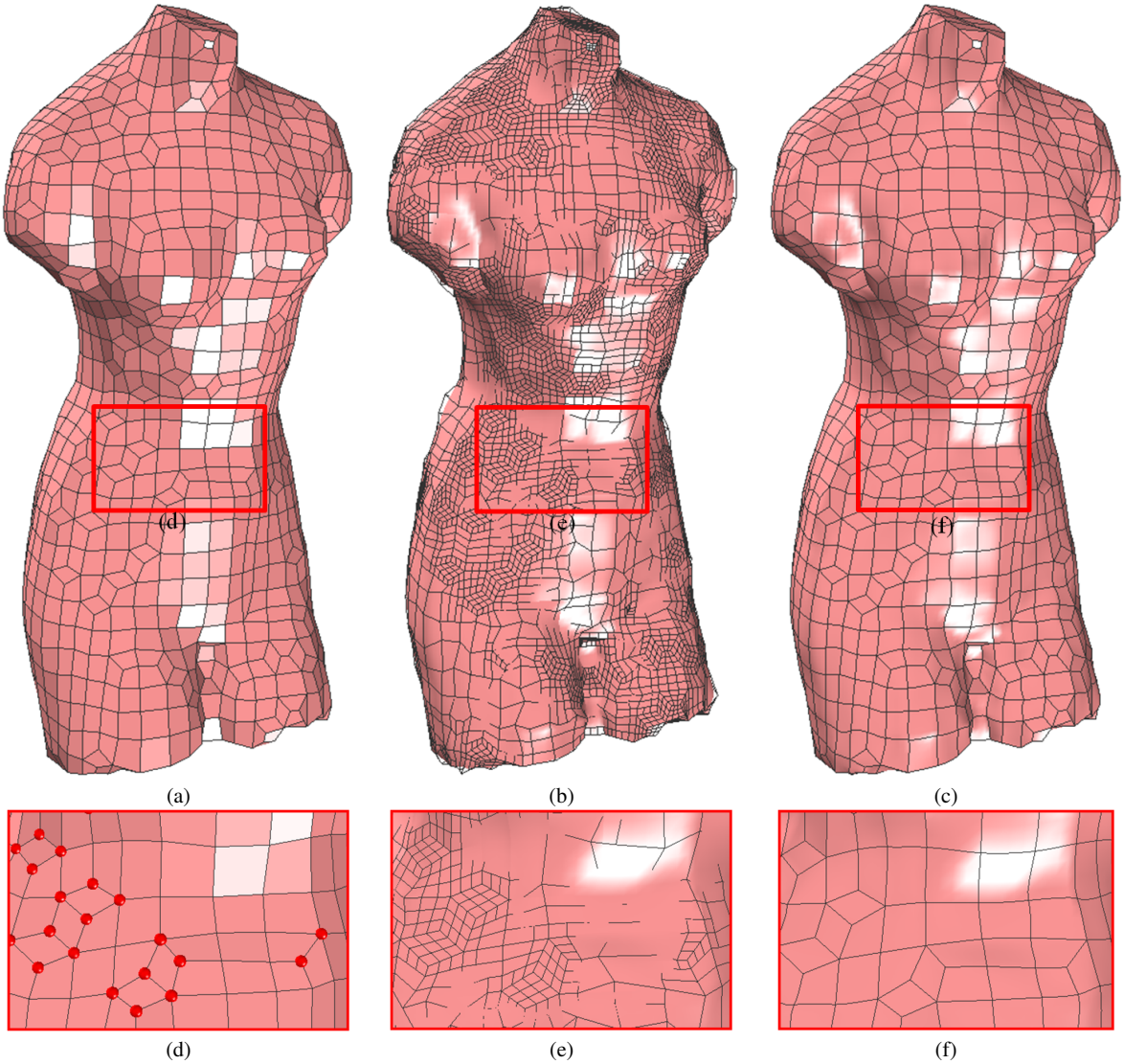


Fig. 24 The venus model. (a) The input unstructured quadrilateral mesh; (b) the constructed T-spline surface and T-mesh; and (c) the extracted Bézier elements. (d) to (f) show details, and red points in (d) denote extraordinary points.

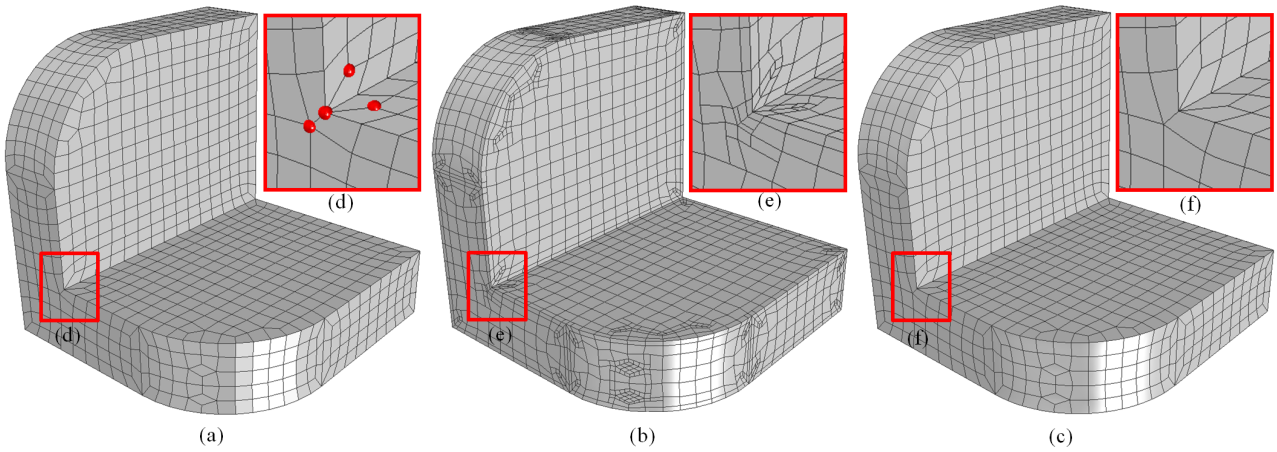


Fig. 25 The multiaxis model. (a) The input unstructured quadrilateral mesh; (b) the constructed T-spline surface and T-mesh; and (c) the extracted Bézier elements. (d) to (f) show details, and red points in (d) denote extraordinary points.

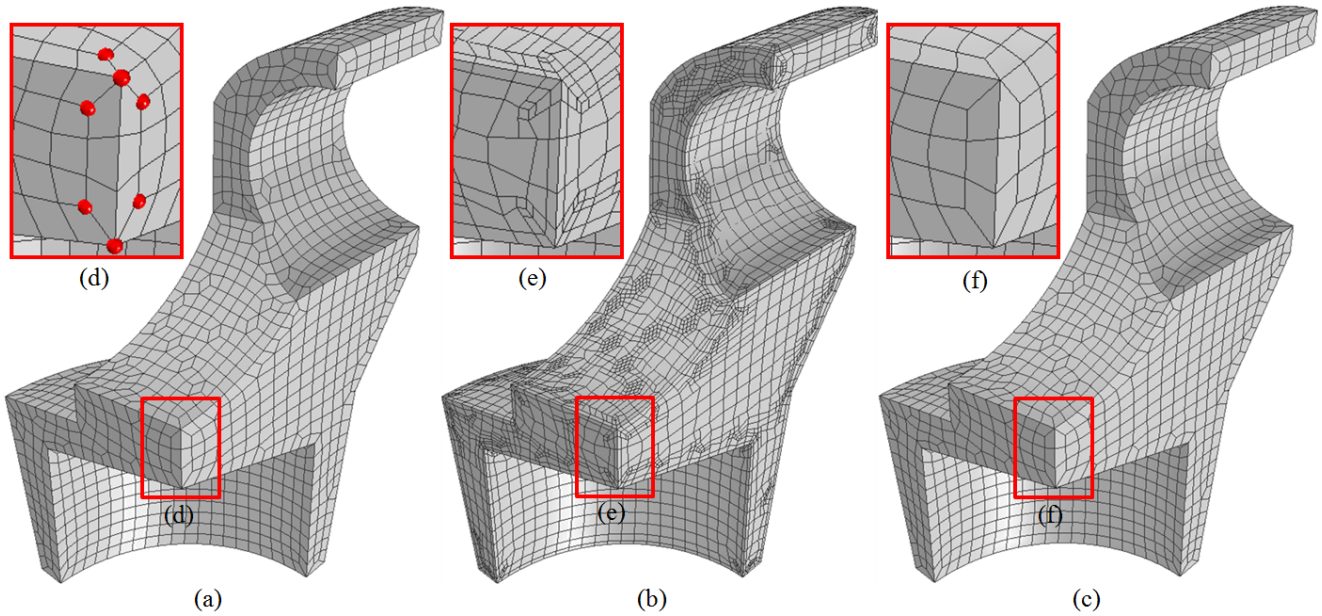


Fig. 26 The hook model. (a) The input unstructured quadrilateral mesh; (b) the constructed T-spline surface and T-mesh; and (c) the extracted Bézier elements. (d) to (f) show details, and red points in (d) denote extraordinary points.

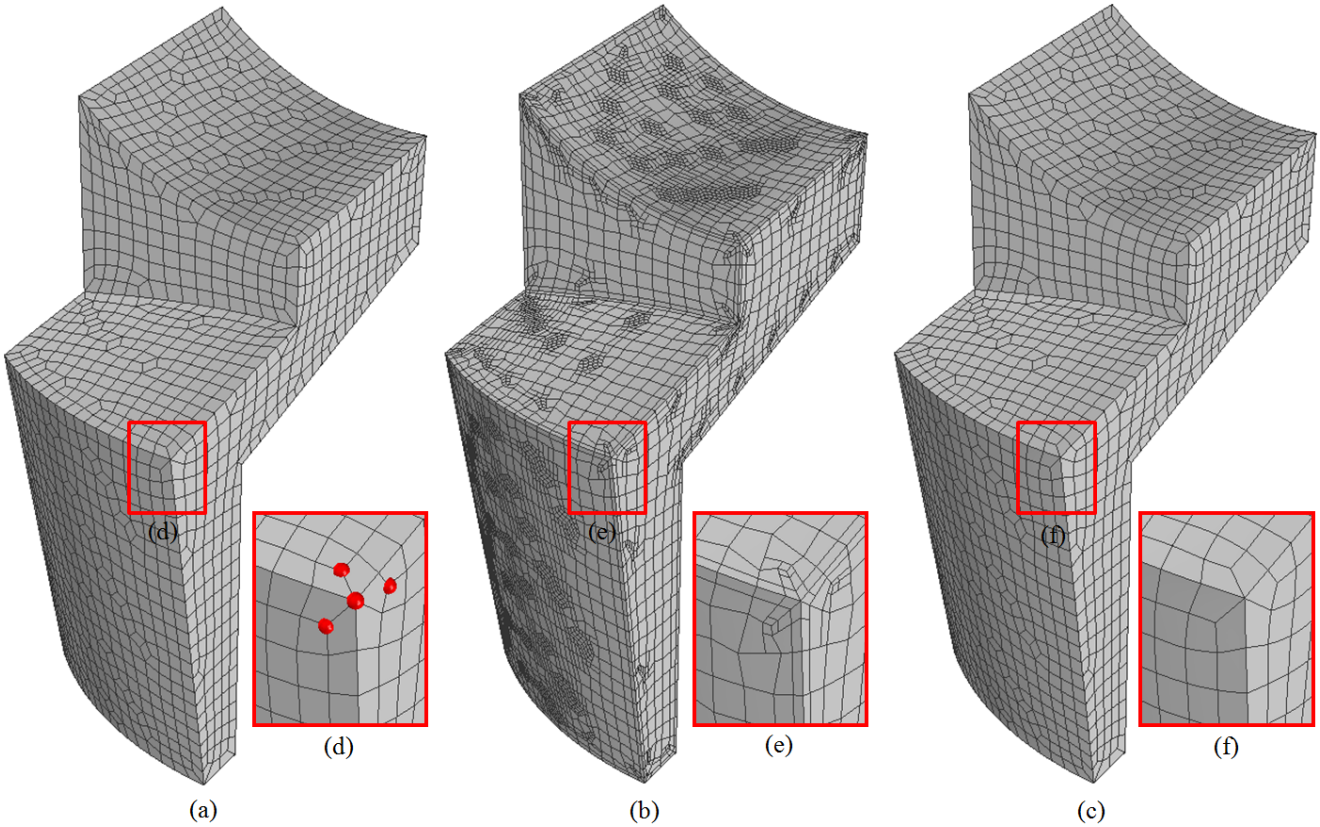


Fig. 27 The hook2 model. (a) The input unstructured quadrilateral mesh; (b) the constructed T-spline surface and T-mesh; and (c) the extracted Bézier elements. (d) to (f) show details, and red points in (d) denote extraordinary points.

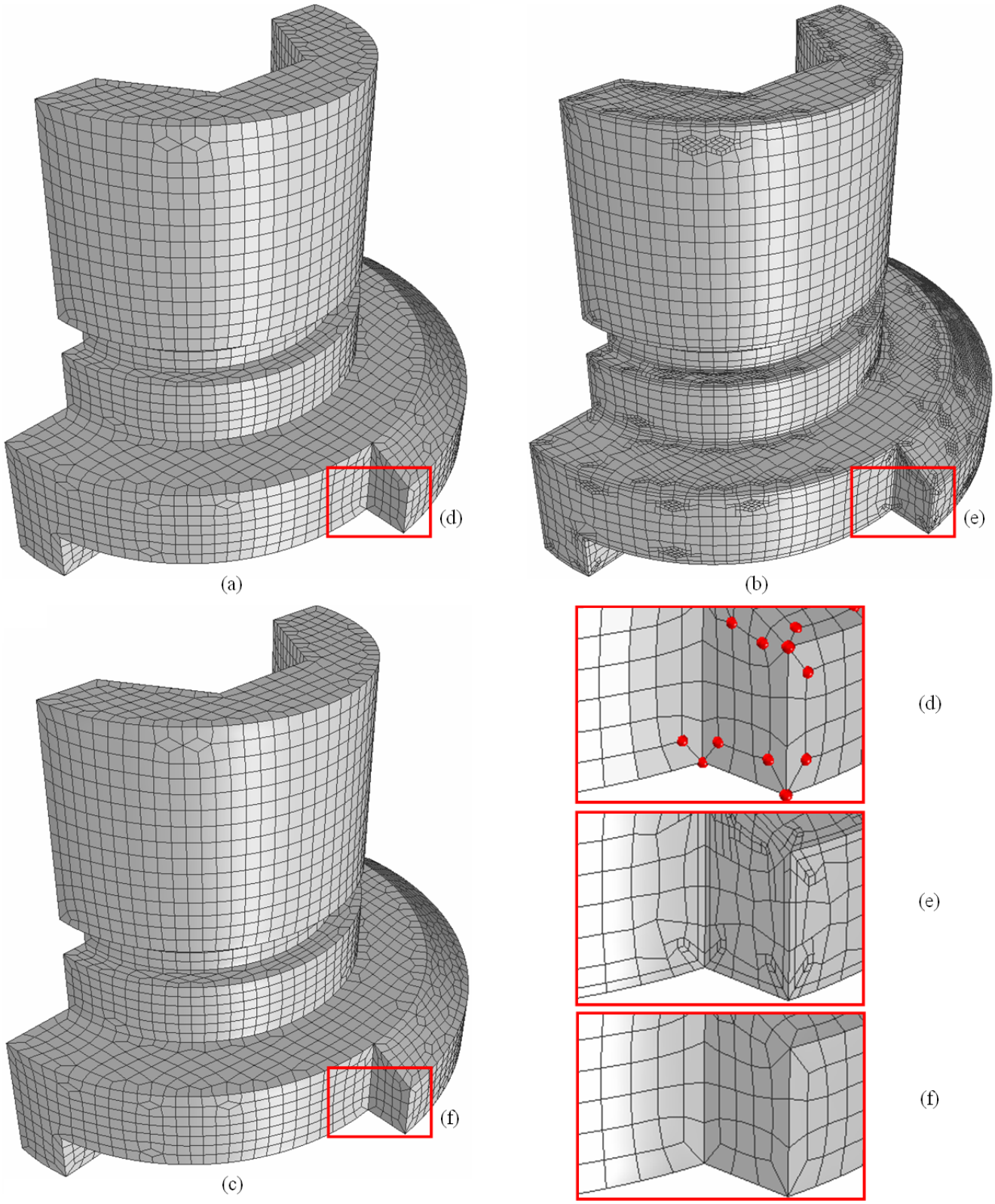


Fig. 28 The varco model. (a) The input unstructured quadrilateral mesh; (b) the constructed T-spline surface and T-mesh; and (c) the extracted Bézier elements. (d) to (f) show details, and red points in (d) denote extraordinary points.

7. T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
8. W. Li. *Automatic mesh to spline surface conversion*. PhD thesis, INPL, France, Nov 2006.
9. W. Li, N. Ray, and B. Lévy. Automatic and interactive mesh to T-spline conversion. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 191–200. Eurographics Association, 2006.
10. L. A. Piegl and W. Tiller. *The NURBS book*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.
11. J. Qian and Y. Zhang. Sharp feature preservation in octree-based hexahedral mesh generation for CAD assembly models. *19th International Meshing Roundtable*, pages 243–262, 2010.
12. M. A. Scott, M. J. Borden, C.V. Verhoosel, T. W. Sederberg, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. Technical Report ICES Report 10-45, the University of Texas at Austin, 2010.
13. T. Sederberg and J. Zheng. T-splines and T-NURCCs. *ACM Transactions on Graphics*, 22(3):477–484, 2003.
14. T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. In *ACM SIGGRAPH 2004*, pages 276–283, 2004.
15. T.W. Sederberg, G.T. Finnigan, X. Li, H. Lin, and H. Ipson. Watertight trimmed NURBS. *ACM Transactions on Graphics*, 27(3), 2008.
16. H. Wang, Y. He, X. Li, X. Gu, and H. Qin. Polycube splines. In *Symposium on Solid and Physical Modeling*, pages 241–251, 2007.
17. Y. Wang and J. Zheng. Control point removal algorithms for T-spline surfaces. *Geometric Modeling and Processing 2006*, 4077:385–396, 2006.
18. Y. Zhang and C. Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering*, 195(9-12):942–960, 2006.
19. Y. Zhang, C. Bajaj, and B.-S. Sohn. 3D finite element meshing from imaging data. *Computer Methods in Applied Mechanics and Engineering*, 194(48-49):5083–5106, 2005.
20. Y. Zhang, Y. Bazilevs, S. Goswami, C. Bajaj, and T.J.R. Hughes. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196(29-30):2943–2959, 2007.